

A Turning Point For Atari?
Report From The Winter Consumer Electronics Show

COMPUTE!

\$2.95
April
1986
Issue 71
Vol. 8, No. 4

\$3.75 Canada
02195
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

**New Technologies:
The Converging Digital
Universe**
Why Everything From
Computers & CD-ROMs
To TVs & Satellites
Are On A Collision Course

**Adding Power
To ST BASIC**
Valuable Techniques
For Atari Programmers

Apple Hi-Res Graphics
Mix, Flip, And Copy
High-Resolution Screens

64 Screen Genie
A Magic Method
For Character Graphics

AmigaDOS Batch Files
Take Control
Of Your Amiga

Atari FontMaker
Design Your Own
Custom Characters



(No I.D. required for half-elves.)

When the Going Gets Tough, the Bard Goes Drinking.

And the going is tough in Skara Brae town. The evil wizard Mangar has cast an eternal winter spell. Monsters control the streets and dungeons beneath. Good citizens fear for their lives. What's worse, there's only one tavern left that serves wine. But the Bard knows no fear. With his trusty harp and a few rowdy minstrel songs he claims



are magic, the Bard is ready to boogie. All he needs is a band of loyal followers: a light-fingered rogue to find secret doors, a couple of fighters to bash heads, a conjurer to create weird allies, a magician for magic armor. Then it's off to combat, as soon as the Bard finishes one more verse. Now what's a word that rhymes with "dead ogre?"



4 classes of magic user, including wizard and sorcerer. 85 new magic spells in all.



128 color monsters, many animated. All challenging.



Full-color scrolling dungeons. 16 levels, each better than the one before. 3-D city, too.

Optional use of
characters created
in Wizardry® or
Ultima III®

The Bard's Tale™

from



ELECTRONIC ARTS™

Speech: 100% machine language, 400K words on 2 disks. 64K RAM minimum, enhanced for 128K. Now available for Apple II family for \$44.95, C-64 & 128 for \$39.95. To order: Visit your retailer or call (800) 227-6705 (In Calif. call (800) 652-7979) for VISA or MasterCard orders. To purchase by mail, send check or money order to Electronic Arts, P.O. Box 306, Half Moon Bay, CA 94019. Add \$3 for insured shipping and handling. Allow 1-4 weeks for delivery. The Bard's Tale is worth the wait! Apple is a trademark of Apple Computer. Wizardry is a registered trademark of Sir-Tech Software, Inc. Ultima is a registered trademark of Richard Garriott. The Bard's Tale and Electronic Arts are trademarks of Electronic Arts. For a copy of our complete catalogue and direct order form, send \$50 and a stamped, self-addressed envelope to Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403.

We don't care which computer you own. We'll help you get the most out of it.



CompuServe puts a world of information, communications, and entertainment at your fingertips.

CompuServe is the world's largest information service designed for the personal computer user and managed by the communications professionals who provide business information services to over one quarter of the FORTUNE 500 companies.

Subscribers get a wealth of useful, profitable or just plain interesting information like national news wires, home

shopping and banking, travel and sophisticated financial data. Plus electronic mail, national bulletin boards, forums (special interest groups), and a multi-channel CB simulator.

You get games and entertainment, too. Board, parlor, sports, space and educational games. Trivia and the first online TV-style game show played for real prizes.

To buy a CompuServe Subscription Kit,

see your nearest computer dealer. To receive our informative brochure or to order direct call or write:

CompuServe®

Consumer Information Service, P. O. Box 20212
5000 Arlington Centre Blvd., Columbus, OH 43220
800-848-8199 In Ohio Call 614-457-0802

An H&R Block Company

Free software from Electronic Arts.™

It's easy!

Buy any of these 12 smash hits from your participating dealer between February 1, 1986, and May 31, 1986. Then just mail in your proof of purchase, \$5 for shipping and handling, and the official "Get One Free" coupon. So what are you waiting for?

In case of emergency...

If you can't find a participating dealer, you can order direct from us by telephone or mail. Just call toll-free 800-227-6703. In California call 800-632-7979. Have your Visa or MasterCard number ready. Just tell us what you want to buy, and what you want for free. It's simple. And there's a 14-day "satisfaction or your money back" guarantee on all products ordered direct.

In case your participating dealer runs out of "Get One Free" coupons, use this.

cut here

☐ Yes! I want free software from Electronic Arts!

☐ No thanks. I'm using my computer as a doorstop.

(Check one)

Apple II, II+, //c, //e

- ☐ Dr. J. & Larry Bird
Go One-On-One
- ☐ Archon
- ☐ Financial Cookbook
- ☐ Music Construction Set
- ☐ Pinball Construction Set
- ☐ Seven Cities of Gold
- ☐ Archon II: Adept
- ☐ Movie Maker

Commodore 64 & 128

- ☐ Dr. J. & Larry Bird
Go One-On-One
- ☐ Archon
- ☐ Financial Cookbook
- ☐ Music Construction Set
- ☐ Pinball Construction Set
- ☐ Seven Cities of Gold
- ☐ Archon II: Adept
- ☐ Movie Maker
- ☐ M.U.L.E.
- ☐ Realm of Impossibility
- ☐ Mail Order Monsters
- ☐ Racing Destruction Set

Atari 400-1200 series

- ☐ Dr. J. & Larry Bird
Go One-On-One
- ☐ Archon
- ☐ Financial Cookbook
- ☐ Music Construction Set
- ☐ Pinball Construction Set
- ☐ Seven Cities of Gold
- ☐ Archon II: Adept
- ☐ Movie Maker
- ☐ M.U.L.E.
- ☐ Realm of Impossibility

IBM PCjr, PC, & comp.

- ☐ Dr. J. & Larry Bird
Go One-On-One
- ☐ Archon
- ☐ Financial Cookbook
- ☐ Music Construction Set
- ☐ Pinball Construction Set
- ☐ Seven Cities of Gold

Please send my free software to the following address. I have enclosed the required Proofs of Purchase and \$5 (check or money order payable to Electronic Arts) to cover the cost of shipping and handling (\$7 Canadian).

PLEASE PRINT

Name

Address

City State Zip

Phone () - -

Signature

Proof of Purchase: Send us the following two items: 1) the dated cash register tape or receipt showing the product you purchased, and 2) find the Command Summary or Reference Card inside the package you purchased. Cut off the part number on the back (lower-left-hand side of the card) or the product name from the front of the card. Send these two items and \$5 for shipping and handling (\$7 Canadian) along with the coupon in **Electronic Arts, "Buy 1, Get 1 Free!"**, P.O. Box 7530, San Mateo, CA 94403. This offer expires May 31, 1986. Open to residents of the U.S.A. and Canada only. Only the products listed are available for this promotion. No substitutions will be accepted.

(no dogs included...)

Let's face it. Buying home software can be a risky proposition. Good software costs a lot of money. Cheap software practically rolls over and barks. But look at this. The titles listed below are Electronic Arts' greatest hits. They've all scored on the "top twenty" best-seller charts. Between them they've won over 60 product quality awards — making them the most honored products in the home software industry.

Best of all, there's no longer any reason to swallow hard when you think about treating yourself to quality software. Because now you can:

Buy 1, Get 1 Free.



DR. J & LARRY BIRD GO ONE-ON-ONE™

The best-selling computer sports simulation of all time.
"Game of the Year" — Electronic Games
Apple II family, C-64 & 128, Atari, IBM PCjr, PC, and comp.



PINBALL CONSTRUCTION SET™

Build your own video pinball games.
"Best Game Generator" — InfoWorld
Apple II family, C-64 & 128, Atari, IBM PCjr, PC, and comp.



MUSIC CONSTRUCTION SET™

Composition for anyone who can pour a penny.
"Best Music Generator" — InfoWorld
Apple II family, C-64 & 128, Atari, IBM PCjr, PC, and comp.



RACING DESTRUCTION SET™

Land mine, oil slicks, flaming spin-outs, collisions and crashes — the ultimate racing game.
— Computer Entertainment
C-64 & 128



SEVEN CITIES OF GOLD™

Play Questador in this educational adventure of history and geography.
"Best Role-Playing Adventure" — Family Computing
Apple II family, C-64 & 128, Atari, IBM PCjr, PC, and comp.



ARCHON™

Like chess with arcade battle scenes.
"Game of the Year" — Creative Computing
Apple II family, C-64 & 128, Atari, IBM PCjr, PC, and comp.



ARCHON II: ADEPT™

Graduate school for Archon addicts. Even more strategy and magic.
"Best Strategy/Arcade Game" — Family Computing
Apple II family, C-64 & 128, Atari



MAIL ORDER MONSTERS™

New! Like an "Archon Construction Set".
"Best Construction Set" — Family Computing
C-64 & 128



MOVIE MAKER™

Create your own high-quality animated movies.
"A masterpiece" — Creative Computing
Apple II family, C-64 & 128, Atari



REALM OF IMPOSSIBILITY™

Action and adventure in a world of Escher-like illusions. Includes a unique 2-player cooperative mode.
C-64 & 128, Atari



M.U.L.E.™

The best multi-player computer game of all time.
"Best Strategy Game" — InfoWorld
"Best Multi-Player Game" — Electronic Games
C-64 & 128, Atari



FINANCIAL COOKBOOK™

Take control of your personal finances. Like a calculator spreadsheet, and financial advice combined.
Apple II family, C-64 & 128, Atari, IBM PCjr, PC, and comp.

Remember — this offer is good only on these 12 products for the Apple II family, C-64 and 128, Atari, and IBM PC, PCjr, and compatible computers. No substitutions will be accepted. For details, see the accompanying coupon.

For a complete Electronic Arts Catalog, send \$24 and a stamped, self-addressed envelope to Electronic Arts Catalog, 1633 Gateway Drive, San Mateo, CA 94404. IBM is a registered trademark of International Business Machines Corp. Apple II is a registered trademark of Apple Computer Inc. Commodore 64 & 128 are registered trademarks of Commodore Business Machines, Inc. Atari is a registered trademark of Atari Computer Corp. Electronic Arts, Dr. J & Larry Bird Go One on One, Archon, Financial Cookbook, Music Construction Set, Pinball Construction Set, Seven Cities of Gold, Archon II — Adept, Movie Maker, M.U.L.E., Realm of Impossibility, Mail Order Monsters, and Racing Destruction Set are registered trademarks of Electronic Arts.

EA
ELECTRONIC ARTS™



Elementary Amiga BASIC
C. Regena
0-87455-041-6, \$14.95
Disk \$15.95



Elementary ST BASIC
C. Regena
0-87455-034-3, \$14.95
Disk \$15.95

BASIC programming at its best!

Two new
programming
guides from
COMPUTE! Books.

Written by the author of the bestselling *Programmer's Reference Guide to the TI-99/4A*, these books introduce you to the new and powerful BASIC on the Amiga and Atari ST personal computers. The computers' impressive graphics, animation, and sound can be unlocked with the right commands, and BASIC is the place to start. Regena shows you how—in the clear, concise language that's made her such a popular writer.

Complete descriptions of the Amiga's and ST's BASIC commands, syntax, and organization take you from novice to full-power programming. Sample programs and subroutines, all ready to type in, are included. Plus, both books offer you working software while showing you how to write your own programs. A disk is also available for each book which includes all the programs from the book in an easy, ready-to-load format.

SPECIAL COMBINATION OFFER Order the book and disk together for only \$29.95!

You'll find these new programming guides and many more useful, entertaining COMPUTE! books at your local computer and book stores. Or you can order directly from COMPUTE! Books.

For the fastest service, call toll free 1-800-346-6767 (in NY 212-887-8525). Or mail the attached coupon with your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please add shipping and handling charges to all orders: In U.S., \$2.00 per book, disk, or combination package; \$5.00 per item airmail.

I want the best in BASIC programming guides. Please send me:

- ☐ Elementary ST BASIC, (034-3), \$14.95 each
☐ Elementary ST BASIC Disk, \$15.95 each
☐ Elementary ST BASIC disk and book combination, \$29.95
☐ Elementary Amiga BASIC, (041-6), \$14.95 each
☐ Elementary Amiga BASIC Disk, \$15.95 each
☐ Elementary Amiga BASIC disk and book combination, (057-2), \$29.95

ALL ORDERS
MUST BE
PAID IN
U.S. FUNDS

Subtotal _____
 NC residents add 4.5% sales tax _____
 Shipping charges
 (\$2.00 per item in U.S. and surface mail,
 \$5.00 per item airmail)
 Total amount enclosed _____

- ☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Account No. _____ Exp. Date _____ (Required)

Name _____

Address _____

City _____

State _____ Zip _____

Allow 4-6 weeks for delivery.

36412112

COMPUTE! Publications, Inc. 

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 9th Floor, New York, NY 10019
*Member of COMPUTE! CONSUMER LIAISON: COMPUTE! is a Reader's Digest COMPANY. BASIC and COMPUTE! are registered trademarks.

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

COMPUTE!

APRIL 1986
VOLUME 8
NUMBER 4
ISSUE 71

FEATURES

- | | | |
|----|--|-----------------|
| 20 | New Technologies: The Converging Digital Universe | Selby Bateman |
| 30 | A Turning Point for Atari?
Report from the Winter Consumer Electronics Show | Tom R. Halfhill |
| 42 | Tug-o-War | Mark Tuttle |

GUIDE TO ARTICLES AND PROGRAMS

64/128/AT/AP/AM/
PC/PCI/ST/TI

REVIEWS

- | | | |
|----|-------------------------------|---------------|
| 51 | Silent Service | Neil Randall |
| 52 | DeluxePaint for Amiga | Lee Noel |
| 53 | 5More for Commodore 64 | Art Hunkins |
| 58 | Heart of Africa | Neil Randall |
| 65 | Hacker | Todd Helmrick |
| | | |
| 65 | MasterType's Writer for Apple | Stephen Levy |
| 66 | HoboWriter for the Atari ST | George Miller |

64/128/AP/AT/PC/PCjr
AM
64
64/128/AP/AT
64/128/AP/AT/
PC/PCjr/AM/ST/Mac
AP
ST

COLUMNS AND DEPARTMENTS

- | | | |
|-----|---|-------------------------------------|
| 4 | The Editor's Notes | Tom R. Halfhill |
| 10 | Readers' Feedback | The Editors and Readers of COMPUTE! |
| 19 | HOTWIRE | |
| 110 | The World Inside the Computer: The Robot Inside You | Fred D'Ignazio |
| 111 | Computers and Society:
Humanizing the User Interface, Part 2 | David D. Thornburg |
| 112 | The Beginner's Page: More String-Slicing | Tom R. Halfhill |
| 113 | Telecomputing Today: An April Trade Show Report | Arlan R. Levitan |
| 114 | Programming the TI: Creating Rhythms | C. Regena |
| 116 | IBM Personal Computing: Two Checkers and a Manager | Donald B. Trivette |
| 117 | INSIGHT: Atari-Binary Files, Unitel | Bill Wilkinson |

•
•
•
•

•
•
•

TI
PC/PCj
AT

THE JOURNAL

- | | | |
|-----|--|--------------------|
| 68 | Loading and Linking Commodore Programs, Part 2 | Jim Butterfield |
| 71 | Adding System Power to ST BASIC, Part 1 | Kevin Myktyyn |
| 74 | Moustify Your Applesoft Programs, Part 2 | Lee Swaboda |
| 79 | IBM Variable Snapshot | Tony Roberts |
| 82 | Atari Textdump | Ralph Johnson |
| 83 | AmigaDOS Batch Files | Charles Brannon |
| 86 | Apple Disk Duper | Jason Coleman |
| 87 | Smooth-Scrolling Billboards for IBM | Paul W. Carlson |
| 89 | Commodore 64 Screen Genie | James A. Ledger |
| 94 | Screen Saver 64 | Stephen E. Masters |
| 97 | Atari FontMaker | Charles Brannon |
| 102 | Hi-Res Graphics Aid Routines | Jon Hylands |
| 104 | Commodore 64 Key Phantom | Melvin Baker |
| 107 | Screen Clock for IBM | Marc Svaliyama |

64/128/VIC/+4/16
ST
AP
PC/PCjr
AT
AM
AP
PC/PCjr
64/128
64/128
AT
AP
64/128
PC/PCjr

109 CAPUTEI Modifications or Corrections to Previous Articles

- 119 News & Products
121 MLX: Mochine Language Entry Program for Atari
123 MLX: Mochine Language Entry Program for Commodore 64
125 COMPUTE!'s Guide to Typing In Programs
128 Advertisers Index

NOTE: See page 125 before typing in programs.

AP Apple, Mac Macintosh, AT
Atari, ST, Atari ST, V VEO-20, 64
Commodore 64, +4 Commodore
Plus/4, 16 Commodore 16, 128
Commodore 128, P PET/CBM, T
Texas Instruments, PC IBM PC, PC
IBM PC, AM Amiga, *General
Interest

TOLL FREE Subscription Order Line
800-247-5470 (In IA, 800-532-1272)

COMPUTE! Publications, Inc.

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., 825 7th Ave., New York, NY 10019 U.S.A. Phone (212) 263-8360. Editorial Offices are located at 324 West Winderover Avenue, Greensboro, NC 27408. Domestic Subscriptions 12 issues, \$24. POSTMASTER: Send address changes to **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50309. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright ©1986 by COMPUTE! Publications, Inc. All rights reserved. ISSN 0194-387X.

Editor's Notes

COMPUTE! Editor Tom Halfhill contributes an editorial this month.

—Robert Lock, Editor in Chief

The latest-generation personal computers present the best evidence to date that microcomputers are evolving into desktop mainframes. Megabytes of memory, hard disk drives, high-speed processors, and multitasking operating systems are no longer limited to the monster machines locked away in the data processing departments of governments and big corporations. Now you can get these features in a personal computer that costs less than \$2,000 and fits comfortably on a desktop.

But why would you want to? After all, many people are questioning why anyone needs any kind of computer in their home. Are the new machines just a more blatant example of technological overkill?

Practically everyone who's ever used a computer understands the value of more memory and mass storage, high-speed processing, and faster input/output. All those things translate into more horsepower, and if the price is right, we'll welcome more horsepower. But one feature that some people are regarding with skepticism is multitasking—the ability to run more than one program at a time. Is it really practical to run a spreadsheet and a word processor simultaneously? Even if the computer can do two things at once, the user probably can't.

This criticism overlooks several advantages of multitasking: its convenience, the way it shifts busy work away from the user and onto the computer, its implications for software design, and its future applications in tomorrow's homes.

It's hard to appreciate the sheer convenience of multitasking until you've experienced it. Even if you aren't actively using two or more programs at once, you can keep them loaded in memory, available at the press of a key or click of a mouse button. For instance, you can type a letter with a word processor, switch to a terminal program to upload it to an electronic mail service, then switch to BASIC to finish a program you've been writing. On most home computers, that would

require rebooting the machine several times, swapping disks, running different programs, and waiting.

Multitasking can also spare you some drudgery by letting the computer do the tedious jobs. If you log onto a commercial information service to check stock quotations every evening, you can set up the computer to do this for you automatically—even while you're using the machine for something else in the meantime. Multitasking is something that's hard to do without once you've had it.

Multitasking also lets you create your own integrated software packages. You can buy whatever word processor, graphics program, spreadsheet, and terminal program you want and load them all into memory at once. If the computer supports a standardized file transfer protocol—as do the Macintosh and Amiga—you can cut and paste pictures or spreadsheet tables into documents created with the word processor and so forth, even if the programs were made by different software companies.

Finally, there are exciting possibilities for multitasking in the future. Remember that microcomputers are following the paths established by mainframes; it's a small step from multitasking to multiuser processing. If a computer can run several programs at once in windows on a single screen, why not turn those windows into separate screens and put them in different rooms? We know from our mail and readership surveys that many of you are already multicomputer households. Mom and Dad have a computer in the study, and the kids have one or two in the family room or bedroom. Someday you'll be able to buy a single personal computer with enough brute force to drive several terminals throughout your home. Each terminal will be as powerful and seemingly as independent as today's personal computers, yet the system will be economical because you'll all share the same printer, modem, hard disk drive, and CD-ROM player.

The main disadvantages of multitasking—the amounts of memory and processing time it can gobble up—are temporary annoyances. Memory chips are getting cheaper as fast as microprocessors are growing more powerful.

Atari recently introduced the first 1024K computer for under \$1,000, just five years after an 8K Atari 800 retailed for \$1,000. And Motorola recently announced a 20 megahertz version of its 68020 microprocessor, referred to as the "mainframe on a chip."

It seems that the only real problem to be overcome is the incredible complexity of writing and debugging a true multitasking operating system. Ask an IBM owner about all the popular Sidekick-type, co-resident programs that compete for the attention of DOS interrupts and the keyboard. Or ask an Amiga owner about the weird things that can happen when the computer tries to do too much at once. (In fact, one of the strangest things we've seen on the Amiga is something that can be described as a "half crash." On practically any other computer, a system crash is a system crash—the machine locks up and you have no choice but to reboot. On the Amiga, we've managed to crash part of the computer while the other part struggles valiantly onward. You end up rebooting anyway just to play safe, but it's an interesting demonstration of multitasking.)

Essentially, multitasking gives you the near-equivalent of several computers in a single box. And if the box is priced right and meets your other requirements, why walk when you can run?

Tom R. Halfhill, Editor

IF YOU CAN FIND A BETTER PROGRAM WE'LL BUY IT FOR YOU!*



WORD WRITER with **Spell Checker** **Now with 85,000 word Spelling Checker**

- An efficient, 80-column professional word processing system which includes a spelling checker and built-in calculator.
- Contains all the features you'll need for everyday word processing, plus most of the sophisticated features found in more expensive programs: document chaining, form letter printout, page separations, horizontal and vertical scrolling, and more.

With Timeworks you get more power for your dollar

You can use each program alone. Or interface this trio — one at a time if you like — into a completely integrated productivity system that delivers all the power and features most of you will ever need . . . at a cost that lets you enjoy their use.

Look for these and other Timeworks programs at your favorite dealer. Or contact Timeworks, 444 Lake Cook Road, Deerfield, IL 60015. Phone: (312) 948-9200.

DATA MANAGER 2

Faster, more efficient, more versatile

- A complete general information storage and retrieval system with report-writing, graphics, statistics, and label-making capabilities.
- Exclusive X-SEARCH, X-SORT, and X-CHART features allow you to cross-search any category of information; sort items alphabetically, numerically, or by date; break down statistical information into categories; and graphically view your results.

With Timeworks you get more than software

You Get Our Customer Technical Support Team

At the other end of our toll-free hotline, you'll find our full-time Customer Technical Support Team. Free of charge to all registered users.

You Get Our Liberal Trade-Up Policy

You'll find the details inside each package.

SWIFTCALC with **SIDEWAYS**

New easy-to-use spreadsheet for home and small business use

- The SIDEWAYS option lets you print all your columns on one, continuous sheet . . . sideways.
- 250 rows and up to 250 columns (128K version) provide up to 62,500 cells (locations on the spreadsheet) in which to place information.
- Performs mathematical functions up to 17 digits. Allows the use of minimum and maximum values, averages, sums, integers, absolute values and exponential notation.

***With Timeworks you get our Money Back Guarantee**

If you can find anything that works better for you, simply send us your Timeworks program, your paid receipt, and the name of the program you want, along with your check or credit card number for any retail price difference. If it's available, we'll buy it for you.**

**For Apple,
Commodore 128 (128K)
& Commodore 64
Computers**



More power for your dollar.

Other Timeworks Programs:

- The Evelyn Wood Dynamic Reader
- Sylvia Porter's Personal Finance Series
- Swiftax ■ Cave of the Word Wizard
- Business Systems ■ Wall Street
- The Electronic Checkbook
- The Money Manager

**These programs interface
with each other
Now available for
Commodore 128**

** Offer valid for 90 days from date of purchase.

*** Registered trademarks of Apple Computer, Inc., International Business Machines Corporation, and Commodore Computer Systems.

© 1985 Timeworks, Inc. All rights reserved.

Publisher James A. Casella
Managing Editor Robert C. Lock
Senior Editor Richard Mansfield
Managing Editor Kathleen Martinek
Executive Editor Sally Solomon

Editor Tom R. Hartzel
Assistant Editor Philip Nelson
Production Director Tony Roberts
Production Editor Gail Cowper
Editor, COMPUTE! GAZETTE Lance Ego
Technical Editor Ditis R. Cowper
Assistant Technical Editor John Krause, George Miller
Program Editor Charles Brannon

Assistant Editor, COMPUTE! GAZETTE Todd Helmarck
Assistant Features Editor Kathy Vokal
Programming Supervisor Patrick Parish
Editorial Programmers Kim Victor, Kevin Mykyryn
Research/Copy Editor Joan Souleau
Copy Editor Ann Davies
Submissions Reviewer Mark Tuttle
Programming Assistants David Florence, David Hersley

Executive Assistant Deb Nash
Administrative Assistants Julia Fleming, Iris Brooks, Mary Hunt, Sallie Apea
Associate Editors Jim Butterfield, Toronto, Canada
 Harvey Herman, Greensboro, NC
 Fred Dignato, Roseland, NJ
 David Thompson, Los Altos, CA
 Bill Wilson

Contributing Editor Bill Wilson

COMPUTE!'s Book Division
Editor Stephen Levy
Assistant Editor Gregg Kiser
Director, Book Sales & Marketing Steve Voyatzis

Production Manager Irma Swan
Art & Design Director James R. Fry
Assistant Editor, Art & Design Lee Noel
Mechanical Art Supervisor De Forner
Artists Dorothy Gray, Catherine Kellrow
Typesetting Terry Cash, Carole Dunton
Illustrator Harry Blair

Director of Advertising Sales Ken Woodard
Production Coordinator Kathleen Horton

Promotion Assistant Caroline Dark

Customer Service Manager Diane Longo
Dealer Sales Supervisor Orchi Tamayo
Individual Order Supervisor Judy Taylor
Receptionist Anita Amfield
Warehouse Manager John Williams

Data Processing Manager Leon Stokes

James A. Casella, President
R. Steven Vetter, Vice President, Finance and Planning

COMPUTE! Publications, Inc. publishes



COMPUTE! Books

COMPUTE!'s GAZETTE DISK

COMPUTE!'s Apple Applications Special

Editorial offices: 324 West Wendover Avenue, Suite 200, Greensboro, NC 27408-3434

Corporate offices: 625 7th Avenue, New York, NY 10019, 212-265-8260

Customer Service: 800-345-6767 (in NY 212-857-5525)

Coming In Future Issues

Two New Columns!
For Atari ST
And Amiga

Hickory Dickory Dock:
Teaching Kids To Tell Time
For Commodore, Atari, ST,
Amiga, Apple, PC/PCjr

Commodore Auto-Boot Maker
Custom Characters
For Atari SpeedScript

Add System Power
To Atari ST BASIC, Part 2

Amiga Puzzle:
An Amiga BASIC Application
Windows On The Commodore 128



Association of Broadcasters



Regular Publishers Association

Subscription Orders

COMPUTE!
P.O. Box 10954
Des Moines, IA 50340

TOLL FREE Subscription Order Line

800-247-5470
in IA 800-532-1272

COMPUTE! Subscription Rates
(12 Issue Year):

US (one yr.) \$24
 (two yrs.) \$45
 (three yrs.) \$65
Canada and Foreign
Surface Mail \$30
Foreign Air
Delivery \$65

Advertising Sales



1. New England
 John Savai
 Eastern Advertising Manager
 212-315-1665

2. Mid Atlantic
 John Savai
 Eastern Advertising Manager
 212-315-1665

3. Southeast & Foreign
 Harry Blair
 919-275-9809

4. Midwest
 Gordon Benson
 312-362-1821

5. Northwest/Mountain/Texas
 Phoebe Thompson
 Dan Nunes
 408-354-5553

6. Southwest
 Ed Winchell
 213-378-8361

Director of Advertising Sales
 Ken Woodard

COMPUTE! Home Office 212-887-8450

Address all advertising materials to:
 Kathleen Horton
 Advertising Production Coordinator
COMPUTE! Magazine
 324 West Wendover Avenue
 Suite 200
 Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 10954, Des Moines, IA 50340. Include a note indicating your preference to receive only your subscription.

Authors of manuscripts warrant that all materials submitted to COMPUTE! are original materials with full ownership rights resident in said authors. By submitting articles to COMPUTE!, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of COMPUTE! Publications, Inc. No portion of this magazine may be reproduced in any form without written permission from the publisher. Entire contents copyright © 1985. COMPUTE! Publications, Inc. rights to programs developed and submitted by authors are explained in our author contract. Unsolicited materials not accepted for publication in COMPUTE! will be returned if author provides a self-addressed, stamped envelope. Programs (on tape or disk) must accompany each submission. Printed listings are optional, but helpful. Articles should be furnished as typed copy (upper- and lowercase, please) with double spacing. Each page of your article should bear the title of the article, date and name of the author. COMPUTE! assumes no liability for errors in articles or advertisements. Options expressed by authors are not necessarily those of COMPUTE!.

RET. IBM, VIC-20 and Commodore are trademarks of Commodore Business Machines Inc. and/or Commodore Electronics Limited. Apple is a trademark of Apple Computer Company. BWP, PC and PCjr are trademarks of International Business Machines, Inc.

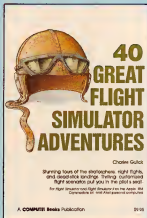
AT&T is a trademark of AT&T Inc. 1-999AA is a trademark of Texas Instruments Inc. Radio Shack Color Computer is a trademark of Radio Shack.

Discover the thrill of flying!



with COMPUTE! Books'
40 Great Flight Simulator Adventures

- Maneuver around the towers of the World Trade Center.
- Sightsee the Hudson River.
- Practice night flying and aerobatics.



40 Great Flight Simulator Adventures

Charles Gulick

Forty exciting, customized flight simulator scenarios put you in the pilot's seat as you fly over bridges, around skyscrapers, and land at mysterious airports. *Flight Simulator* (IBM PC) and *Flight Simulator II* (Apple II, Commodore 64, Atari) are two of the most popular games/simulations for personal computers. With this book, you can experience flight adventures from the moment you load the program. Parameters set up each flight and a running commentary describes what you'll see (and where to look to see it). Ranging from the simple and straightforward to the advanced and even mystical, these 40 flights will open a new dimension to an already outstanding program.

\$9.95 ISBN 0-87455-022-X

Note: *Flight Simulator* from Microsoft Corporation and *Flight Simulator II* from Sublogic Corp. are required in order to use this book. The book is designed to enhance the programs.

Please send me _____ copies of *40 Great Flight Simulator Adventures* at \$9.95 each.
(ISBN No. 0-87455-022-X)

All orders must be prepaid in U.S. funds.

Subtotal _____
NC residents add 4.5% tax _____
\$2.00 shipping and handling
charge per book. _____
Total amount enclosed _____

To order this exciting adventure guide, mail
the attached coupon with your payment to
COMPUTE! Books, P.O. Box 5038,
F.D.R. Station, New York, NY 10150. Or call
toll free 1-800-346-6767 (in NY 212-887-8525.)

- ☐ Payment enclosed (check or money order)
☐ Charge ☐ Visa ☐ MasterCard ☐ American Express

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4-6 weeks for delivery.

36432211

COMPUTE! Publications, Inc. 
One of the ABC Publishing Companies

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.



Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

Relocating Machine Language

I would like to combine two Commodore machine language programs that both reside at location 49152 (\$C000). I know that BASIC lets you relocate programs quite easily, just by moving the bottom-of-BASIC pointer upward. How is this done with ML programs?

Richard Sands

Machine language programs written for a 6802-based computer are usually quite difficult to relocate. For instance, say that you have an ML program at \$C000 which starts with these instructions:

```
LDA $C030,X
JSR $C200
JMP $C400
```

None of these instructions can be relocated unless you change the address contained in the instruction itself. The first (LDA \$C030,X) retrieves one byte of data from a table beginning at location \$C030 (note that the data lies within the program code). The JSR instruction works like GOSUB in BASIC, so JSR \$C200 goes to a subroutine located at \$C200 and then returns. JMP works like GOTO in BASIC. JMP \$C400 sends the computer straight to the segment of code located at \$C400. Now say that you move the entire program down to location \$8000. The instruction JSR \$C200 still sends the computer to \$C200, but that address isn't within the program any more. To make the code work correctly at \$8000, you'd have to change these three instructions to the following:

```
LDA $8030,X
JSR $8200
JMP $8400
```

That's not particularly difficult, and some machine language monitors even have a special command to make such adjustments automatically. However, you must be careful not to change addresses

that refer to locations outside the program:

```
JSR $FFD2
```

This instruction calls the standard Commodore print-a-character routine, located in the computer's ROM. If you mistakenly adjust this address along with all the internal address references, the result may be disastrous. Now let's look at a more difficult case:

```
LDA ($FB),Y
```

This instruction uses the powerful and very common indirect Y addressing mode, which refers indirectly to an address held in two successive zero page addresses (locations \$FB-\$FC in this case). There's no way to tell by looking at this instruction alone whether it refers to an area inside the program (and hence requires adjustment) or something external to the program code (in which case adjustment may be a mistake). You'll have to disassemble the program in its entirety, looking for other instructions that affect the contents of locations \$FB-\$FC, either directly or indirectly. If this instruction is part of a general-purpose subroutine, you may find that it's called by many different parts of the program. Since free zero-page space is limited, you may also find that other subroutines re-use locations \$FB-\$FC for an entirely different purpose. And while it's obvious that an instruction like STA \$FB affects the contents of \$FB, what about ROR \$03,X or STA (\$B0),Y? Those instructions might just as easily change the address held in \$FB-\$FC.

Once you've sorted out all the indirect addressing, you'll need to check for self-modifying routines—code that changes its own instructions while it runs. When that's done, you'll have to interpret all the program's data and variable areas. For instance, say that you find the following hexadecimal values in a memory dump of the program code:

```
93 05 20 C4 54 0D 41 43
```

These bytes could be virtually anything—sprite shape data, characters for a printed message, part of an internal dispatch table, preset values for a bunch of unrelated variables, or even garbage that will be replaced with something meaningful when the program runs. While some programmers locate data areas at the end of the program, others sprinkle data and

variables freely throughout the code. Until you find out exactly what purpose these bytes serve, there's no way to tell whether they need adjustment. This problem, more than any other, makes it impossible to write an "automatic ML relocater" that works correctly in every case. The relocater would need to have as much intelligence as a knowledgeable ML programmer who thoroughly understands the subject program.

These problems generally don't apply to 68000-based computers like the Amiga, Atari 520ST, and Macintosh. Since the computer normally decides for itself where to load the ML code, most 68000 ML programs must be relocatable. That's no great hardship for programmers, since the 68000 instruction set includes many relocatable instructions.

128 Atari Colors

Here is a machine language program that allows your Atari computer to display 128 colors at the same time. The program displays a different color on each horizontal display line.

```
10 FOR I=0 TO 21:READ A:P
   OKE 1536+I,A:NEXT I
20 POKE 752,1:PRINT CHR$(
   125):A=USR(1536)
30 DATA 173,11,212,201,32
   ,208,249,141,10,212,14
   ,24,208,232,232,208,2
   ,44,142
```

```
40 DATA 24,208,248,232
```

David Boyer

Thank you for the example.

Using Preview-80 With 64 SpeedCalc

I own a Commodore 64 and look forward to getting new programs from COMPUTE! each month, especially utilities. After typing in SpeedCalc (COMPUTE!, January 1985), I was pleased to find that the "Preview-80" program (COMPUTE! GAZETTE, November 1985) works just as well with SpeedCalc as it does with SpeedScript. This lets you preview a SpeedCalc file in 80-column format on the screen before printing it out. The procedure for using Preview-80 is the same as usual. First, load Preview-80 with LOAD "PREVIEW80",8.1. Then type NEW, and load SpeedCalc as you would normally. Instead of typing

WINTER GAMES. THE QUEST FOR THE GOLD CONTINUES...



You've captured the gold in Summer Games® and Summer Games II™. Now it's on to the Winter Games! And what an incredible setting—a completely realistic winter wonderland featuring seven action-packed events.

At the Ski Jump you control your form in mid-air, knees straight, leaning forward. Hot Dog Aerials challenges your courage and your sense of humor. In Figure Skating you leap into Double and Triple Lutz jumps—wow the crowd with a perfect Camel into a Sit Spin. It's timing and style that counts. Free Skating lets you choreograph your own routines. In Speed Skating it's you against a fellow speed demon—the fastest human beings on level earth! And the Bobsled—still faster as you fly around hairpin turns, leaning hard to stay in the tube. Finally the Biathlon, the ultimate challenge to your endurance in cross-country skiing and marksmanship.

All of this fun and excitement is easy to learn and play. You control the

action with the joystick, animating your player for style and rhythm. You choose the country you want to represent. Listen to its national anthem. Then it's practice, training and learning a winning strategy for each event. Now the Opening Ceremony and the competition begins—against your friends or the computer. Will you be the one who takes the gold at the Awards Ceremony? Will your name be etched amongst the World Record holders?

The quest for the gold continues... And it's all here—the strategy, the challenge, the competition, and pageantry of Winter Games!

APPLE MAC G4/128
Winter Games



EPYX
COMPUTER SOFTWARE

1043 Kiel Ct., Sunnyvale, CA 94089



Strategy Games for the Action-Game Player®

RUN to start SpeedCalc, type SYS52000 and press RETURN. SpeedCalc will become active as usual. But when you press SHIFT-CTRL-P for printed output, and then press S for output to the screen, Preview-80 takes over. All of the Preview-80 options are available; to exit the Preview-80 window, press RUN/STOP twice.

Bob Starr

Thanks for the tip.

Moving AmigaDOS Commands to RAM

Regarding your article "Introduction to AmigaDOS" (COMPUTE!, January 1985), I feel that it's inconvenient to have all of the AmigaDOS commands—especially often-used commands like DIR—stored on disk rather than in memory. Is it possible to load all or part of DOS into RAM? If so, how much memory does it take up? Can you write a batch file to make this part of the boot sequence? Will AmigaDOS become RAM-resident in the future?

Barry Silverstein

Every AmigaDOS command is disk-resident, and you're not likely to see any change in the near future. This can be inconvenient at times since, for each separate AmigaDOS command, the computer has to access the same Workbench disk that was present when you booted the system. If you have only one disk drive, this scheme creates delays and requires extra disk-swapping. Fortunately, there's a simple remedy. If you create a RAM disk, you can then COPY any or all of the AmigaDOS commands from floppy disk to RAM disk; the amount of memory consumed depends on how many commands you copy. Once that's done, an ASSIGN command tells the system to use the RAM-resident commands.

The most convenient way to move AmigaDOS commands into RAM is by editing the startup-sequence file, which is similar to an AUTOEXEC.BAT file in PC/DOS and MS/DOS systems. When you insert a disk in response to the Amiga's Workbench disk prompt, the computer looks in the S subdirectory of the currently mounted disk for a file named startup-sequence. If this file is present, the computer executes the AmigaDOS commands that it contains. Since startup-sequence is an ordinary ASCII text file, it's easy to modify with a word processor or any text editor that handles ASCII files. (Before editing this file, make sure that you have at least one copy of the Workbench disk in addition to the one that came with your computer.) If you edit this file with Textcraft or some other word processor, you must restate it in the form of plain ASCII text, without special formatting characters or control codes.

AmigaDOS includes two text editors of its own. The easiest one to use is called ED. Type this line at the CLI prompt, then press RETURN:

```
ed "s/startup-sequence"
```

This command activates ED and loads startup-sequence into the editor. An unmodified startup-sequence file looks like this:

```
"Workbench disk. Version 1.1"
ECHO " "
ECHO "Use Preferences tool to set date."
ECHO " "
LoadWb
endcli > nil;
```

You'll probably recognize the messages that appear on the screen when you boot up with that disk. The LoadWb command loads and activates the Workbench, and endcli terminates the AmigaDOS command sequence, returning you to the Workbench screen. We'll use ED to add some new command lines between LoadWb and endcli. ED is a very simple text editor; Use the cursor keys to move around in the file, and the BACKSPACE key to delete characters. Everything that you type is inserted at the current cursor position (you can use uppercase if you like, but lowercase works just as well and is easier to type).

While you could copy the entire command directory (named C) into the RAM disk, that wastes a lot of RAM since some AmigaDOS commands are used only rarely. To save memory, we'll copy only the most commonly used commands. Place the cursor on top of the E in endcli and enter these lines, pressing RETURN at the end of each line:

```
echo "Copying AmigaDOS commands to
RAM disk..."
copy c/copy ramc/copy
assign x: ramc/copy
assign d: ramc
cd sysc
x assign d:
x cd d:
x copy d:
x delete d:
x dir d:
x diskcopy d:
x echo d:
x ed d:
x endcli d:
x info d:
x list d:
x mkdir d:
x newcli d:
x rename d:
x run d:
x type d:
cd sys:
assign x: ramc
assign d: cdelete
```

Remember, this set of commands goes between the LoadWb and endcli lines in the normal startup-sequence file. If you change your mind and don't want to modify the file, press ESC-Q followed by RE-

TURN; ED returns you to the CLI without changing anything. To save the modified file to disk, press ESC-X followed by RETURN. After the file is resaved, ED returns you to the CLI prompt. To test the new startup-sequence file, reboot the computer by pressing CTRL-Left Amiga-Right Amiga. It takes about a minute to copy the commands shown above. Once the process is finished, all of the copied commands are instantly available in RAM (if this doesn't work, reload startup-sequence into ED and check for typing mistakes).

The first command line following ECHO copies the COPY command itself into RAM so the computer can copy subsequent commands without accessing the disk each time. The next three lines simplify your typing job: The first ASSIGN command tells the computer to substitute the characters ramc/copy wherever it sees the characters x:. The second ASSIGN creates another short alias (d) which stands for the pathname ramc. The CD command changes the current directory to SYS:C so you won't need to specify a subdirectory for every file you want to move. These three shortcuts let you abbreviate all of the remaining COPY commands (the command x: endcli d: becomes the equivalent of ramc/copy sysc/endcli ramc, and so on).

Thus, each line beginning with x: causes the computer to copy a single AmigaDOS command to the RAM disk. Of course, you can delete commands from this list, or add others if desired. The command ASSIGN C: RAM:C tells the computer to use the C directory in the RAM disk as its command directory. From this point on, the Amiga searches the RAM disk when you tell it to execute an AmigaDOS command. The final ASSIGN command isn't really necessary, but shows how to create a shorthand name for an often-used command. In this case, we're creating d: as a synonym for DELETE. Once this is done, you can delete the file TEST by typing either DELETE TEST or D:TEST. This can be done for any command, using whatever shorthand you like. The command sequence shown here is adapted from an example in COMPUTE!'s AmigaDOS Reference Guide, which explains this and many other AmigaDOS topics in detail.

HELP For Atari XL And XE

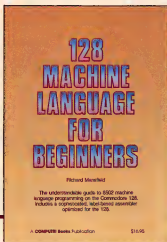
I have an Atari 800XL and would like to know how to read the HELP key.

R.E. Brock

The status of the HELP key can be determined by PEEKing location 732 on the Atari XL and XE computers. If the HELP key alone is pressed, this location returns a value of 17; when SHIFT and HELP are pressed simultaneously, it contains 81.

TAP THE POWER of the Commodore 128

By the author of
*Machine Language
for Beginners* and
*Second Book of
Machine Language*



128 Machine Language for Beginners

Richard Mansfield

One of the bestselling computer books ever has now been completely revised for the Commodore 128. Most commercial software is written in machine language because it's far faster and more versatile than BASIC. This new edition of *Machine Language for Beginners* is a step-by-step introduction to 8502 machine language programming on Commodore's 128 computer.

The book includes everything you need to learn to effectively program the 128: numerous programming examples, memory management tutorials; a complete description of the many Kernal routines and other new 128 features; numerous hints and programming techniques; and a dictionary of all major BASIC commands and their machine language equivalents. It also includes a high-speed, professional-quality, label-based assembler, optimized to take advantage of the speed and extra memory of the 128.

0-87455-033-5

\$16.95

Like the other top-quality books from COMPUTE!, *128 Machine Language for Beginners* brings you ready-to-use information in a clear, lively style that makes learning easy and enjoyable, whether you are a beginner or an advanced computer user.

An optional disk is also available which includes the assembler and example programs in the book. The *128 LADS Disk* is fully tested and ready to load on the Commodore 128. It costs only \$12.95 and saves you hours of typing time.

Order your copy of *128 Machine Language for Beginners* and the *LADS Disk* today. Call toll free 1-800-346-6767 (in NY 1-212-887-8525) or mail your payment (plus \$2.00 shipping per book or disk) to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE! Publications, Inc. 

One of the ABC Publishing Companies
825 7th Avenue, 8th Floor, New York, NY 10019
Publishers of COMPUTE! (Copyright © 1986), COMPUTE! (Copyright © 1987), and COMPUTE! (Copyright © 1988) and COMPUTE! (Copyright © 1989) and COMPUTE! (Copyright © 1990) and COMPUTE! (Copyright © 1991) and COMPUTE! (Copyright © 1992) and COMPUTE! (Copyright © 1993) and COMPUTE! (Copyright © 1994) and COMPUTE! (Copyright © 1995) and COMPUTE! (Copyright © 1996) and COMPUTE! (Copyright © 1997) and COMPUTE! (Copyright © 1998) and COMPUTE! (Copyright © 1999) and COMPUTE! (Copyright © 2000) and COMPUTE! (Copyright © 2001) and COMPUTE! (Copyright © 2002) and COMPUTE! (Copyright © 2003) and COMPUTE! (Copyright © 2004) and COMPUTE! (Copyright © 2005) and COMPUTE! (Copyright © 2006) and COMPUTE! (Copyright © 2007) and COMPUTE! (Copyright © 2008) and COMPUTE! (Copyright © 2009) and COMPUTE! (Copyright © 2010) and COMPUTE! (Copyright © 2011) and COMPUTE! (Copyright © 2012) and COMPUTE! (Copyright © 2013) and COMPUTE! (Copyright © 2014) and COMPUTE! (Copyright © 2015) and COMPUTE! (Copyright © 2016) and COMPUTE! (Copyright © 2017) and COMPUTE! (Copyright © 2018) and COMPUTE! (Copyright © 2019) and COMPUTE! (Copyright © 2020) and COMPUTE! (Copyright © 2021) and COMPUTE! (Copyright © 2022) and COMPUTE! (Copyright © 2023) and COMPUTE! (Copyright © 2024) and COMPUTE! (Copyright © 2025)

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from Holt, Rinehart, & Winston, 55 Horner Avenue, Toronto, ON M8Z 4X8.

THE CMO ADVANTAGE

- ✓ THE BEST PRICES!
- ✓ Next day shipping on all in stock items
- ✓ Free easy access order inquiry
- ✓ Orders from outside Pennsylvania save state sales tax
- ✓ Free technical support with our factory trained technical staff
- ✓ There is no limit and no deposit on C.O.D. orders
- ✓ There's no extra charge for using your MasterCard or Visa. Your card is not charged until we ship
- ✓ No waiting period for cashiers checks
- ✓ We accept purchase orders from qualified corporations. Subject to approval
- ✓ Educational discounts available to qualified institutions
- ✓ FREE CATALOG MEMBERSHIP

ORDER LINE

CALL TOLL-FREE
1-800-233-8950
 Educational Institutions
 Call Toll-Free
1-800-221-4283
CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A204

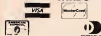
MAILING ADDRESS

Computer Mail Order
Dept. A204
 477 East Third Street
 Williamsport, PA 17701



MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS



SHIPPING

Add 3%, minimum \$7.00 shipping and handling on all orders. Larger shipments may require additional charges

All items subject to availability and price change

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
 Ontario/Quebec

1-800-268-4559
 Other Provinces

1-416-828-0866
 in Toronto

TELEX: 06-218960

2505 Dunwin Drive,
 Mississauga, Ontario
 Canada L5L1T1

All prices shown are for U.S.A. orders.
 Call The Canadian Office for Canadian prices

HOME COMPUTERS

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
No LCD Display	CALL

ATARI

65XE (64K)	CALL
126XE (128K)	CALL
520XE (512K)	CALL
800XL 8K	CALL
1010 Recorder	\$49.99
1050 Disk Drive	CALL
1020 Printer	\$29.99
1027 Letter Quality Printer	\$129.00
1030 Direct Connected Modem	\$59.99

COMMODORE

C128 Computer	\$269.00
C128 Disk Drive for C128	\$49.00
C1282 (RGB 13" Monitor for C128)	CALL
C1630 (Modem for C128)	\$175.00
C64 Computer	CALL
C1541 Disk Drive	\$199.00
C1535 Diskette	\$59.00
C1802 Color Monitor	\$109.00
C1660 Auto Modem	\$59.99
DPS 1101 Dotsy Printer	\$399.00
Comox 220 (C64 Interface)	\$89.99

GRAPHICS

■ Polaroid	
Polaris	\$139.00
Power Processor	\$299.99
Illuminated Slide Mounter	\$39.99
Polaroid 2 Pack film	\$19.99

ACCESSORIES

CURTIS POWER STRIPS	
Diamond SP-1	\$32.99
Emerald SP-2	\$32.99
Sapphire SP-1	\$52.99
Ruby SPF-2	\$39.99
Sale Strip SP-3	\$19.99
KENSINGTON	
MasterPeace	\$99.99
MasterPeace Plus	\$119.00

PORTABLE COMPUTERS

HEWLETT PACKARD	
410CV	\$139.00
410CX	\$199.00
HP 11C	\$49.99
HP 10C	\$75.99
HP 15C	\$75.99
HP 16C	\$83.99
HPIL Module	\$99.00
HPIL Cassette or Printer	\$369.99
Card Reader	\$143.50
Extended Function Module	\$63.99
Time Module	\$63.99

We stock the full line of
 HP calculator products

NEC

PC-6001 LS	CALL
PC-6201 Portable Computer	\$319.00
PC-6201 Disk Drive	\$699.00
PC-6221A Thermal Printers	\$149.00
PC-6201A Data Recorder	\$399.00
PC-6201-06 8K RAM	\$79.99

SHARP

PC-1350	\$149.99
PC-1261	\$149.99
PC-1500A	\$169.00
PC-1255A	\$89.99
CE-125 Printer/Cassette	\$129.00
CE-150 Color Printer Cassette	\$149.00
CE-161 18K RAM	\$129.00

SOFTWARE

ARTWORK

Bridge 4.0 (AI)	\$17.99
Strip Poker (AI)	\$19.99

ATARI

ROMS FOR ATARI ONLY	
6030 Alan Writer	\$24.99
Star Raiders	\$4.99
Missile Command	\$4.99
Defender	\$4.99
Galaxian	\$4.99
Asteroids	\$4.99
Carnage	\$4.99

BROOKERBUND

Print Shop (AI)	\$29.99
Graphics Library I, II, III (AI)	\$17.99
Bark Street Writer (AI/II)	\$34.99
Bark Street Writer (II/III)	\$44.99

BATTERIES INCLUDED

Paperclip (AT, 64/128)	\$37.99
Paperclip (II/III)	\$44.99
Homepack (64/AT)	\$29.99
DEGAS (AI)	\$27.99

CONTINENTAL SOFTWARE

Home Accountant (AI)	\$44.99
Tax Advantage (AI)	\$44.99
Home Acc./Tax Bundle (AI)	\$59.99

MICROPROSE

Kennedy Approach (AI)	\$24.99
Aeropolis (AI)	\$24.99
Stent Series (AI)	\$24.99

PFS

File (64)	\$30.99
File, Report, Graph (II/III)	\$79.99
Wide w/Print (64/II)	\$79.99

SPRINGBOARD

Newroom (64)	\$34.99
Clip Arts (64)	\$22.99
Newroom (Apple)	\$22.99
Clip Arts (Apple)	\$22.99

SUBLOGIC

Flight Simulator (AI)	\$37.99
Jet (64)	\$37.99

DISKETTES

maxell	
3 1/4" 55/100 (10)	\$24.99
3 1/4" 55/100 (10)	\$24.99
5 1/4" MD-1 whetstones (10)	\$34.99
5 1/4" MD-2 whetstones (10)	\$19.99
5 1/4" MD-3 HD for AT (10)	\$39.99
3 1/4" 5 pack 55/100 Case	\$13.99

Verbatim

5 1/4" 55/100	\$12.99
5 1/4" 55/100	\$12.99
Disk Analyzer	\$24.99

Dennison

Elephant 5 1/4" 55/100	\$13.99
Elephant 5 1/4" 55/100	\$14.99
Elephant 5 1/4" 55/100	\$16.99
Elephant Premium 55/100 (54)	\$19.99
Elephant 3 1/4" 55/100	\$29.99

5 1/4" 55/100 floppy disks	
(Box of 10)	\$29.99

DISK HOLDERS

INNOVATIVE CONCEPTS	
Flign File 30	\$2.99
Flign File 50	\$14.99
Flign File 50 wallet	\$19.99
Flign File 100	\$19.99
Flign File Data Case	\$9.99

AMARAY

50 Disk Top 3 1/4"	\$9.99
30 Disk Top 3 1/4"	\$9.99

MODEMS

ANCHOR

Volkmodem	\$59.99
Volkmodem 300/1200	\$169.99
Signalhorn Express	\$252.00
Lightning 2400 Baud	\$399.00
Express	\$159.00
6470 (64/128) 300/1200 Baud	\$139.00

DIGITAL DEVICES

AT300 - 300 Baud (Atari)	\$59.99
--------------------------	---------

Hayes

Smartmodem 300	\$139.00
Smartmodem 1200	\$369.00
Smartmodem 1200B	\$379.00
Smartmodem 2400	\$529.00
Micromodem II	\$149.00
Smart Com II	\$59.99
Chronograph	\$199.00
Transet 1000	\$309.00

AST

Reach 1200 Baud Mail Card	\$399.00
---------------------------	----------

SUPRA

MPP-1064 ADMA (4 Supra)	\$69.99
-------------------------	---------

Novation

Smart Cat Plus	\$259.00
J-Cat	\$99.99
Novation 2400	\$549.00
Apple Cat II	\$229.00
212 Apple Cat II	\$379.00
Apple Cat 212 Upgrade	\$229.00
Macmodem	\$279.00

QUADRAM

Quadmodem II	
300/1200	\$399.00
300/1200/2400	\$499.00

EVEREX

1200 Baud Internal (IBMPC)	\$199.00
----------------------------	----------

DRIVES

HARD

1110H Bernoulli Single II	CALL
A210H Bernoulli Dual 30	CALL
A200H Bernoulli Dual 30	CALL
Save on Bernoulli Carts	CALL

TALOGRADES

25, 35, 50, 80 meg (PC)	
100 meg (PC)	\$1299.00

IRWIN

Tape Backup	CALL
-------------	------

EVEREX

40 Meg Internal Backup System 750.00	
--------------------------------------	--

CORE

AT30-AT72MB	CALL
-------------	------

KITS

40, 80 MB (PC, XT, AT)	CALL
10 Meg with controller	\$379.00
20 Meg with controller	\$519.00

FLOPPY

INDUS

Alan GT	\$199.00
G-64 1/28 GT	\$199.00

TANDEM

S01 C-64 Single	\$219.00
S22 C-64 Dual	\$499.00

320K 5 1/4" (PC)	\$119.95
------------------	----------

TEAC

320K 5 1/4"	\$119.00
-------------	----------

MONITORS

PRINTERS

SOFTWARE FOR IBM

IBM

AMDEK	
Video 300 Green	\$119.00
Video 300A Amber	\$129.00
Video 310A Amber TTL	\$159.00
Color 300 Composite	\$189.00
Color 300 Hi-Res. RGB	\$399.00
Color 710 Ultra Hi-Res	\$439.00
Color 722 Dual Mode	\$529.00

MAGNAVOX	
3562 RGB/Composite	\$279.99

NEC	
JB1205A	\$79.99
JB1275G/1275A (ea)	\$99.99
JB1285G TTL	\$129.00
JB1285A TTL	\$129.00
JC1490 RGB	\$229.00
JC1225 Composite	\$179.00
JC1401 Multi Sync RGB	\$549.00

PRINCETON	
MAX-12E Amber	\$179.00
MAX-9 RGB	\$499.00
HX-9E Enhanced	\$219.00
HX-12 RGB	\$499.00
HX-12E Enhanced	\$599.00
SR-12 Hi-Res	\$599.00
SR-12P Professional	\$699.00

TAXAN	
115 12" Green	\$119.99
115 12" Amber	\$129.00
121 TTL Green	\$139.00
122 TTL Amber	\$149.00
610 510x300 RGB	\$199.00
630 640x300 RGB	\$199.00
640 720x400 RGB	\$199.00

QUADRAM	
8400 Quadchrome II	\$499.00
8410 Quadchrome II	\$599.00
8420 Quadchrome II	\$179.99
8500 Quad Screen	\$1449.00

ZIV	
ZVM 1200 Amber	\$99.99
ZVM 1200 Green	\$99.99
ZVM 1240 IBM Amber	\$149.00
ZVM 135 RGB	\$459.00
ZVM 1330 RGB	\$499.00
ZVM 1360 RGB	CALL
ZVM 1260 E G Comp.	CALL

INTERFACES

AST	
Multi IQ (Apple II)	\$159.00

QUADRAM	
GraphicCard	\$79.99
Serial Card	\$99.99
MicroBuilder II+	\$169.00
MicroBuilder 32K	\$189.00

QUADRAM	
Microfizer	from \$139.00
Elizer (Epson)	from \$79.99

Orange Micro	
Grapple-CD (C64)	\$89.99
Grapple+ (Apple)	\$89.99
Grapple-386 (Apple)	\$159.00

DIGITAL DEVICES	
Ape Face (Atan)	\$49.99
U-Print A (Atan)	\$54.99
U-A16Buffer (Atan)	\$74.99
U-Call Interface (Atan)	\$79.99
U-Print C (C64)	\$49.99
P-16 Print Buffer	\$74.99
U-Print 16 Apple II	\$99.99

Canon	
A40, A50, A55	CALL
LBP-M1 Laser	CALL

CITIZEN	
MSP-10 (80 col)	\$279.00
MSP-16 (132 col)	\$369.00
MSP-20 (80 col)	\$349.00
MSP-25 (132 col)	\$509.00

CITIZEN	
Prowriter 7500	\$169.00
Prowriter 1500P	\$349.00
Smartwriter 10-80	\$399.00
3500 Tr Printer	\$1499.00

CORONA	
Laser LP-300	\$2799.00

DIABLO	
D25 Desktopwheel	\$549.00
635 Desktopwheel	\$699.00
D60F Desktopwheel	CALL

epsonwriter	
2900	\$699.00

EPSON	
Homecenter 10, LX-80	CALL
FX-85, FX-86, FX-100, ZX-80	CALL
DX-10, DX-26, DX-35	CALL
SO-2000, H-80, H5-50, AP-80	CALL
LQ-800, LQ-1000, LQ-1500	CALL

JIUKI	
6000 Letter Quality	CALL
6100 Letter Quality	CALL
6200 Letter Quality	CALL
3300 Letter Quality	CALL
5610 Dot Matrix	CALL

LEONO	
808 Dot Matrix 100 cps	\$179.00
1000 Dot Matrix 100 cps	\$259.00
1360 Dot Matrix 120 cps	\$289.00
1385 Dot Matrix 165 cps	\$339.00

NEC	
3000-Series	\$779.00
3000-Series	\$1099.00
ELF 360	\$399.00
Prowriter 590	\$999.00

OKIDATA	
182, 183, 186, 193, 2410, B4	CALL
Okimate 10 (Specially C64/Atan)	\$189.00
Okimate 20 (IBM)	CALL

Panasonic	
KX1090	NEW
KX1091	\$259.00
KX1092	\$399.00
KX1582	\$469.00
KX1595	\$699.00

QUADRAM	
Quadlet	\$399.00
Quad Laser	CALL

SILVER-REED	
500 Letter Quality	\$279.00
550 Letter Quality	\$419.00
800 Letter Quality	\$699.00

SGE	
SG-10A (Atan)	CALL
SG-16C (C64 interface)	CALL
SBSDS/SGSR Series	CALL
Powerbyte Letter Quality	CALL

Texas Instruments	
T1850	\$599.00
T855	\$639.00
T1855	\$799.00

TOSHIBA	
1340 (80 columns)	\$399.00
P341 (132 columns)	\$799.00
P351 (132 columns)	\$1049.00

AREA SOFTWARE	
Peridot	\$549.00

ASTON-TATE	
Framework II	\$399.00
diase II Plus	\$399.00

BORLAND	
Lightstep	\$54.99
Softbook (unprotected)	\$49.00
Reflex	\$54.99
Newspeak	\$54.00

CENTRAL POINT	
Copy II PC-Backup	\$29.99

DECISION RESOURCE	
Chartmaster	\$229.00
Segmenter	\$189.00
Design Master	\$219.00

FIFTH GENERATION	
Fast Back	\$9.99

FOX AND OELLER	
Quackdox III	\$169.00

FUNK SOFTWARE	
Sideways	\$44.99

HARVARD SOFTWARE INC.	
Total Project Manager	\$269.00

INFOCOM	
Comerstone	\$279.00

LIFETIME	
Volkwriter II	\$159.00

LIVING VIDEOTEXT	
Think Tank	\$109.00
Ready	\$64.99

LOTUS	
Symphony	CALL
I-2-3	CALL

MEGA SOFTWARE	
Managing Your Money 2.0	\$99.99

MICROPRO	
Easy	\$94.99
WordStar 2000	\$299.00
WordStar 2000+	\$299.00
WordStar Professional	\$199.00

MICROPHIN SOFTWARE	
R Base 4000	\$249.00
R Base 5000	\$399.00
Clout 2.0	\$129.00

MICROSOFT	
Flight Simulator	\$34.99
MultiPlan	\$129.00
Word	\$229.00
Mouse	\$139.00

MICROSOFT	
CrossTalk XVI	\$89.99
CrossTalk Mark IV	\$149.00
Remote	\$89.99

MULTIMATE	
Multi Mate Word Proc	\$219.00
Advantage	\$269.00
On File	\$89.99
Juni Wm	\$89.99

NOUROMON	
Instat	\$89.99

NORTON	
Norton Utilities 3.1	\$59.99

ONE STEP	
Golf's Best	\$37.99

PFE: IBM	
Proof	\$59.99
File/Graph	(ea) \$79.99
Report	\$74.99
Write/Proof Combo	\$119.00

PROFESSIONAL SOFTWARE	
Write N-Spell	\$59.99

THE SOFTWARE GROUP	
Enroute	\$329.00

SATELLITE EYEVIEW	
Word Perfect 4.1	\$219.00

SORDINQUE	
Accounting	
APAR/GRIN/INVOICE (ea)	\$299.00
SuperCalc 81	\$109.00
EasyWriter II System	\$139.00
Super Project	\$199.00

EPI SOFTWARE	
Open Access	\$379.00

SUBLOGIC	
Jot	\$37.99

IBM PC SYSTEMS
Continued to your specifications.
Call for Best Price!

IBM PC, IBM XT, IBM AT	
IBM	
PC-158 Series, PC-148 Series, PC-154 Series, PC-156 Series, PC-171 Series, AT 200 Series	CALL

SANYO	
MBC 560-2, MBC 565-2, MBC 675 Portable, MBC775, MBC 560 Desktop	CALL

SONO	
Station 7300	CALL
6300	CALL

CORONA	
PPC260 Dual Portable	\$1269.00
PPC270 10 meg Portable	\$1899.00
PC-0002 Dual Desktop	\$1399.00
PC-0002 HD 10 meg	\$1569.00

ITT	
ITT X TRA	
256K, 2 Drive System	CALL
264K 10 meg Hard Drive System	CALL
XPS, 20 meg	CALL

SPERRY	
Sperry-AT as low as \$1749.00	
Sperry-XT as low as \$2499.00	
Call for Specific Configuration!	
All Models	CALL

KAYPRO	
KP 3000 Portable	CALL
Kaypro PC	CALL

MULTIFUNCTION CARDS	
AST	
Rangeage	\$279.00
See Pack Plus	\$229.00
IO Plus II	\$139.00
Advantage-AT	\$399.00
Graph Package	\$249.00
MicroGraph Plus	\$399.00
Power Mate	\$299.00
PC Net Cards	\$379.99
325 V11 Cable	\$69.00
325 V12 Remotes	\$579.00

dca	
ISMA 3270	\$679.00
ISMA Print	\$999.00
ISMA Smart Act	\$779.00

EVEREX	
Edge Link	\$259.00
Graphics Edge	\$239.00
Magic Card II	\$193.00

HERTZ	
Graphics	\$299.00
Color	\$159.00

IBM America	
IDEA 5251	\$589.00

MYLEX	
The Chairman	\$439.00

PARADISE	
Color/Info Card	\$149.00
Modular Graphics Card	\$259.00
Multi Display Card	\$219.00
Five Pack C, S	\$129.00

PERSYST	
Bob Board	\$359.00

TEGMAR	
Captain - 64	\$199.00
Graphics Master	\$459.00

QUADRAM	
Quadport-AT	\$349.00
Liberty-AT (128K)	\$449.00
The Gold Quadboard	\$449.00
The Silver Quadboard	\$299.00
Expanded Quadboard	\$799.00
Liberty	\$259.00
Quadpoint	\$499.00
QuadLink	\$499.00
QuadColor	\$199.00
Chronograph	\$79.99
Parallel Interface Board	\$54.99

INTEL	
PCNCR587 5MHz	CALL
PCNCR587-2 6 MHz	CALL
1010 PC Above Board	FOR
1110 PG-Above Board	FOR
2010 AT-Above Board	PC

Pressing CONTROL and HELP returns a value of 145. The statement POKE 732,0 clears location 732, so you can check for subsequent keypresses.

Apple IIe/IIc Compatibility

I'm interested in buying an Apple IIc computer. Can it use IIe hardware and software?

Carlos Aguayo

The Apple IIc computer is basically an Apple IIe that has been redesigned to take up as little space as possible. To keep the IIc small, Apple left out the IIe's expansion slots (where additional hardware can be attached), but added a built-in 5¼-inch disk drive. They also put the most common IIe expansion hardware (80-column video display, an extra 64K of memory, and two serial input/output ports) on the main board of the IIc. In addition, the IIc has some features that weren't available when the IIe appeared: an advanced 65C02 microprocessor and a character set called Mousetext which contains extra characters especially for Macintosh-style icons- and menu-based programs. The newest version of the IIc (called Enhanced IIc) does have these extra features; dealers can upgrade an older IIc at a small cost.

The IIc can run almost all IIe programs, as long as no special hardware is required. For instance, some music programs can communicate with instruments through a MIDI (Musical Instrument Digital Interface) adapter. This adapter must connect to an expansion slot, which is possible only on a IIe. Other programs sometimes expect a parallel I/O interface to attach a printer. Since the IIc has only serial I/O, it can't run that type of modification. Although the IIc has no expansion slots, its peripherals (serial ports, disk drives, etc.) act like they are built into certain slots. Apple tried to select the most commonly used slot for each peripheral (printer in slot 1, disk drive in slot 6). However, not everyone puts everything in the same place, and some programs may demand an unconventional configuration. IIe owners can rearrange the cards in their slots to run such programs, but IIc owners don't have this option.

The serial ports on the IIc generate standard RS-232 signals which can be used to communicate with most modems from any manufacturer. Many of the most popular printers are also available with RS-232 interfaces. But the IIc does not have standard connectors for these ports. To save space on the back panel of the computer, DIN-type connectors are used instead; as a result, you'll need special cables (available from Apple dealers) to attach serial peripherals.

When it comes to expandability, the IIc is much more flexible than the IIe. Almost any kind of peripheral can be

attached through one of its slots, including parallel I/O ports, MIDI interfaces, hard disk drives, coprocessors, huge RAM expansion cards, and a host of other devices. However, some third-party companies have begun modifying the IIc to put in extras like additional memory and 280 processors (to run the CP/M operating system, a popular IIe add-on). It's still more difficult than expanding a IIe, but it can be done.

IBM PUT And GET

I own a TI-99/4A and an IBM PCjr. Lately, I've been trying to convert some programs from TI to IBM. I have only one problem: the PUT and GET graphic statements in the IBM system. I really don't understand them. Could you show me a way to make an image and move it?

Billy Mobley

First, be aware that IBM BASIC has two types of GET and PUT statements: one for graphics and another for random files. The syntax for each type is different, so be sure you're using the graphics type. GET grabs the screen image within a specified rectangle and stores a copy of it in an array. PUT does just the opposite, putting the image from an array back onto the screen.

Several important rules apply to PUT and GET. Before using either command, you must be in a graphics mode (SCREEN 1, for example); neither PUT nor GET works on a text screen. The array that you GET a shape into must be a one-dimensional numeric array dimensioned to the proper size. Finally, you must GET before you can PUT.

The most difficult task is deciding what size to dimension the array. If the array is too small, it can't hold the graphics image, and the program won't work. The simplest solution is to try a large size like DIM A(500). It won't hurt to dimension it larger than necessary, but this method wastes memory. Here's a more efficient formula that tells you the minimum required size for the array:

$$\text{INT}((4 + \text{INT}((x * \text{res} + 7) / 8) * y) / \text{prec})$$

In this formula, the variable x represents the width of the image in pixels; y is the height of the image; res is 1 for high resolution and 2 for medium resolution; and prec is the precision of the array (2 for integer, 4 for single precision, and 8 for double precision).

GET must be followed by the screen coordinates of two opposite corners of the rectangular image, and the name of the array. For example, GET (0,0)-(19,29),A grabs a 20 × 30 pixel image at the top-left corner of the screen and stores it in array A. (Of course, you must first have an image on the screen. This can be done with DRAW.) With a high-resolution

screen and a single-precision array, the formula above gives 23, so the dimension statement would be DIM A(23).

PUT is followed by the coordinates of the location on the screen where the top-left corner of the image is placed, then the name of the array, and an optional parameter for special effects. Five special effects are available: PSET, PRESET, AND, OR, and XOR. If no special effect is specified, XOR is assumed.

PSET displays the image exactly as it appeared when GET was used. PRESET displays a negative image. AND displays only those parts of the image that overlap an image already on the screen. OR superimposes the image onto an image already on the screen. XOR is a combination of AND and PRESET, reversing only those parts of the image that overlap an image already on the screen. The best way to understand exactly what these special effects do is to try them yourself. Using our example, PUT (200,100),A,PSET displays the image stored in the A array in the center of the screen.

The operation of XOR may seem strange, but it's handy for animation. When you PUT using XOR twice in the same position, the screen is restored unchanged. This allows you to move an image over a background image, giving a 3-D effect. Animation with XOR is a three-step process: PUT the image on the screen with XOR, calculate the new position, PUT the image in the old position a second time to erase it. By performing these steps repeatedly, the image seems to move. The following program moves a ball across the screen.

```
10 SCREEN 1
20 DIM A(113)
30 CIRCLE (20,20),20
40 PAINT (20,20)
50 GET (0,0)-(40,40),A
60 CLS
70 FOR C=1 TO 100
80 PUT (X1,Y1),A 'display image
90 X2=X1+1:Y2=Y1+1 'calculate new position
100 PUT (X1,Y1),A 'erase image
110 X1=X2:Y1=Y2 'old=new
120 NEXT 'repeat
```

Simpler Absent Printer Test

I'm writing with regard to the "Readers' Feedback" item on absent Commodore printers, published in the December 1985 COMPUTE!. Another way to avoid a DEVICE NOT PRESENT error is to access the appropriate device (4) through the command channel (15) and check the value of the status variable ST. If ST does not equal 0, then the printer is not present. Here is a short routine to demonstrate:

```
10 OPEN 15,4,15:CLOSE 15
20 IF ST<>8 THEN 40
```

All the exciting, entertaining, and educational games, applications, and utilities from **COMPUTE!** magazine are now available on disk

for your Commodore,
Atari, Apple, or IBM
personal computer.

The **COMPUTE!** Disk

A new *COMPUTE! Disk* is published every month, rotating among the four major machines covered by *COMPUTE!*: Commodore 64 and 128; Atari 400/800, XL, and XE; Apple II-series; and IBM PC, PCjr, and compatibles.

Every three months you can receive a disk with all the quality programs from the previous three issues of *COMPUTE!* that will run on your brand of computer.

Like the popular *COMPUTE!'s Gazette Disk*, the *COMPUTE! Disk* is ready-to-load and error-free. It saves you valuable hours of typing time and eliminates typing errors.

With a subscription, you will receive one disk every three months for a total of four disks a year—for only \$39.95. That saves you \$20 a year off the single-issue cost.

Or you can order individual issues of the *Disk* for \$12.95 a disk plus \$2.00 shipping and handling.

Remember to specify your type of computer when ordering the *COMPUTE! Disk*. You'll find more information about this month's *COMPUTE! Disk* in this issue. (Note: You'll need the corresponding issues of *COMPUTE!* magazine to use the *Disk* since the disk will have no documentation.)

For fastest service when ordering a subscription to the *COMPUTE! Disk*, call toll free 1-800-247-5470 (in Iowa 1-800-532-1272).

For more details or to order individual issues of the *COMPUTE! Disk*, call our Customer Service Department toll free at 1-800-346-6767 (in New York 212-887-8525).

Please allow 4-6 weeks after placing an order for your first disk to arrive.

COMPUTE! Publications, Inc. 

One of the ABC Publishing Companies
825 7th Avenue, 8th Floor, New York, NY 10019
Publishers of *COMPUTE!* *COMPUTE!'s Gazette* *COMPUTE! Disk* and *COMPUTE! Apple Applications*

The 1050 DUPLICATOR IS HERE...

THE 1050 DUPLICATOR: The most powerful diskdrive copy system ever developed for the ATARI.

The Duplicator for the New "ST" is now available

The only Copy System You will ever need!

What will it do?

► **The main purpose of the Duplicator is to copy disks!** You will be able to copy just about any disk! The copier you make will run on any Atari drive. The Duplicator need not be present to turn your backup copies. The Duplicator is fully automatic. You need only insert source and destination disks. Custom formats will be read and in turn reproduced on the backup copy disk. Our device will reproduce any custom format or heavily copy guarded schemes, bad sectors, double sectors, 19 through 24 sector format will present no problem to the Duplicator.

► **You will still have single density, density and one half, and double density.** When you have a Duplicator installed in a 1050 drive that drive will be turned into true double density. You will have twice the disk storage. Your drive will be compatible with other double density drives at The Sims Int'l. Percom, etc.

HARDWARE POWER

Fully Compatible with the XL & New XE Series.

► **High speed read & write.** Your disk drive will read and load all of your software, saving wear and tear on your drive. The 1050 drive now reads one sector at a time. This is slow and inefficient. With the duplicator installed you will be able to read eighteen sectors in the time it takes standard, unenhanced drives to read one.

► **Included with every Duplicator will be user friendly disk software.** A simple, menu driven program will allow you to copy all of your software. A Duplicator enhanced drive will be a SMART drive. We plan to write many new and exciting programs that can only be run on an enhanced drive, eg. sending a copy-guarded disk over the phone. Since the drive is now fully programmable, future upgrades can be made available to you on disks, should the need arise. No further hardware changes will ever be needed. The Duplicator comes with a full hardware and software guarantee.

\$149⁹⁵

Only

Plus \$2.50 for shipping handling
Add 7% outside U.S.A.

N.Y. State Residents add 7% Sales Tax
Dealer inquiries are welcome. Call for quantity price quote

EASY 5 MINUTE INSTALLATION

NO HARM TO YOUR DRIVE OR INCOMPATIBILITY PROBLEMS CAN EVER ARISE AS A RESULT OF THE INSTALLATION OF OUR DUPLICATOR

IMPORTANT: Only a hardware device like the DUPLICATOR can backup heavily copy-guarded disks. Don't be fooled by software programs that claim to do this.



DUPLICATING TECHNOLOGIES Inc.

Formerly Gardner Computing

99 Jancho Tpke., Suite 302A, Jancho N.Y. 11753

Order between hrs. (516) 333-5805, 5807, 5808

Order by's (516) 333-5950

TERMS: We accept American Express, Visa, MasterCard and C.O.D. orders. Foreign orders must be in U.S. dollars. All personal checks allow 14 days to clear. Most items shipped within 24 hours.



```
30 OPEN 1,4:PRINT#1,"PRINTER I
  S ON":CLOSE 1:END
40 PRINT CHR$(147):"TURN ON PR
  INTER":GOTO 10
```

If you run this program with the printer off, it instructs you to turn the device on. Printing begins as soon as the printer is active.

Jim Plaveccky

Thanks for this compact, all-BASIC solution. In programs that open disk files or use an RS-232 device (usually a modem), you may want to perform this check at the very beginning, before you perform any other OPEN statements. The statement CLOSE 15 closes all other channels in addition to the command channel, terminating any RS-232 communications and disconnecting (but not really closing) any open disk files.

Emphasized TI Character Set

The custom character set given for the Commodore 64 on pages 108-109 of COMPUTE's January 1986 issue can be used on the TI-99/4A with only slight modifications. Since that character definition data is listed in hexadecimal format, it can be read as a pattern-identifier string and assigned with the CALL CHAR statement (see page II-76 in the

TI User's Reference Guide). Each line in the Commodore character set listing contains data for a single character plus a checksum value at the end of the line. To convert the data in each line to a 16-character pattern-identifier string, type in the first eight two-digit hexadecimal numbers (spaces are left out, of course). In the first line, for instance, the resulting string could be used with a CALL CHAR statement to redefine the @ character. To create the new character set, first enter this program:

```
100 FOR L=1 TO 94
110 READ CS
120 CALL CHAR(L+32,CS)
130 PRINT CHR$(L+32);
140 NEXT L
150 GOTO 150
```

Next, you must enter a series of lines containing DATA statements. Each DATA statement represents the data for one character in the form of a 16-character pattern identifier string. For example, the first DATA line would look like this:

```
500 DATA 7CC6DEDEC0C07800
```

Here is how to enter all of the DATA lines.

```
lines 500-800 Use data from line
7108-7178 (defines
```

line 810	ASCII characters 33-63)
lines 820-1070	Use data from line 7000 (ASCII 64)
line 1080	Use data from lines 7206-72D0 (ASCII 65-90)
line 1090	Use data from line 70D8 (ASCII 91)
lines 1100-1110	Use data from lines 00C06030180C0600 as data (ASCII 92)
line 1120	Use data from lines 70E8-70F0 (ASCII 93-94)
line 1130	Use 00000000000000FF as data (ASCII 95)
lines 1140-1390	Use data from line 7200 (ASCII 96)
lines 1400-1420	Use data from lines 7008-70D0 (ASCII 97-122)
line 1430	Enter data from lines 72D8-72E8 (ASCII 123-125)
line 1440	Enter 000020745C080000 as data (ASCII 126)

The result of your effort will be an emphasized font with true lowercase. John Hedstrom

Thank you for your suggestion.

HOTWARE: Software Best Sellers

					Systems				
This Month	Last Month	Title	Publisher	Remarks	Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.		<i>Jet</i>	SubLogic	Jet simulation	•		•		
2.	4.	<i>Silent Service</i>	MicroProse	Submarine simulation	•	•	•	•	
3.	3.	<i>Ultima IV</i>	Origin Systems, Inc.	Fantasy game	•	•	•	•	
4.	2.	<i>Karateka</i>	Brederbund	Action karate game	•	•	•		
5.	1.	<i>F-15 Strike Eagle</i>	MicroProse	Air combat simulation	•	•	•	•	
Education									
1.	1.	<i>Typing Tutor III</i>	Simon & Schuster	Typing instruction program	•		•	•	•
2.	2.	<i>Math Blaster!</i>	Davidson	Introductory math program, ages 6-12	•	•	•	•	•
3.	3.	<i>New Improved MasterType</i>	Scarborough	Typing instruction program	•	•	•	•	•
4.	4.	<i>Music Construction Set</i>	Electronic Arts	Music composition program	•	•	•		
5.	5.	<i>I Am The C-64</i>	Creative/Activision	Introduction to the C-64			•		
Home Management									
1.	1.	<i>Print Shop</i>	Brederbund	Do-it-yourself print shop	•	•	•		
2.		<i>The Newsroom</i>	Springboard	Do-it-yourself newspaper	•		•	•	
3.		<i>Bank Street Writer</i>	Brederbund	Word processor	•	•	•	•	
4.	3.	<i>Print Shop Graphics Library III</i>	Brederbund	Upgraded graphics library	•	•	•		
5.	4.	<i>Print Shop Graphics Library</i>	Brederbund	100 additional graphics	•	•	•		

Copyright 1986 by Billboard Publications, Inc. Compiled by the Billboard Research Department and reprinted by permission. Data as of 2/1/86 (entertainment) and 2/8/86 (education and home management).



IT ALSO RUNS ON 64K.



Serious runners know it takes more than great running shoes to improve performance. It takes knowledge. Now Puma gives you both. With the RS Computer Shoe. The first training shoe to combine advanced footwear technology with computer technology.

The RS Computer Shoe has a custom-designed gate array built into its heel. This computer chip records your run, then communicates the results to any Apple IIe, Commodore 64 or 128, or IBM PC computer.

A software program included with the shoe automatically calculates your time, distance and calories expended. Then graphically compares them to past performances and future goals.

The RS Computer Shoe from Puma. We're so out front in technology, we put computers in the backs of our shoes.



OUR WORD FOR QUALITY

Apple is a registered trademark of Apple Computer, Inc. Commodore 64 and 128 are trademarks of Commodore Computer Systems. IBM and IBM PC are registered trademarks of IBM.



NEW TECHNOLOGIES

The Converging Digital Universe

Selby Bateman, Features Editor

The winds of technological change have been blowing a gale for the past few years. And the forecast shows no indication of a letup. In fact, millions of consumers will begin to reap a resulting whirlwind of new high-tech products for the home, office, and classroom. Consider the following:

- A home stereo system answers your phone, takes messages, and alerts you to incoming calls.
- With the push of a button, your video film recorder captures a picture from your favorite TV show and instantly prints out a still photo for your wallet.
- Your 20-volume set of encyclopedias, contained and cross-indexed on a compact disc in a player connected to your computer, searches and prints out 37 reference sources on your selected topic in less than 30 seconds.
- The satellite dish in your backyard automatically tracks various communication satellites based on the pattern of TV programs you want to watch each night. At the same time, your computer is receiving and storing financial data that unobtrusively shares the same incoming satellite transmission to your TV.
- The digital TV in your living room displays two small windows on the screen while you watch a

The digitization of America is well under way. Thanks to a wave of new consumer electronics products, this year more people than ever will see and hear how the convergence of digital audio, video, satellite, telephone, optical, laser, television, and computer technologies is transforming the world. Yet, the phenomenon is just beginning.

program uninterrupted; one window shows the changing stock quotations, while the second window displays a program from a different channel or previews a tape from your videocassette recorder.

- The computer image recorder connected to your personal computer makes a 35 mm slide, color print, or overhead transparency of the business chart or digital painting you've just created.

Does any of this sound far-fetched? You'll be able to buy products this year that do all of these things and more. If it seems difficult to keep up with the latest news about consumer electronics, it's not your fault. Never have so many dramatic technological changes produced so many new capabilities and products in so short a time. What has become strikingly clear is that all of these innovations share a

common foundation—the digital, microprocessor-based world of computer electronics.

These changes have become so important to our lives and our pocketbooks that market researchers are now targeting a new group of consumers: Technologically Advanced Families (TAFs). Could "yuppies" eventually be surpassed in importance by "taffies," households that purchase and use the latest computers, VCRs, stereo TVs, 8 mm camcorders (camera recorders), compact disc players, satellite dishes, and dozens of other products? Consumer electronics manufacturers and retailers believe that these households are the important leading-edge market for their array of new products.

Among the catalysts sparking enthusiasm for the latest in high-tech gear, none is more important than the personal computer phenomenon of the past half-dozen years. Not only are computer owners the bedrock of the TAFs, but the new generation of 16/32-bit computers is powerful enough to work with just about any other consumer electronics product. Suddenly, devices like VCRs, compact disc players, electronic keyboards, and camcorders have become computer peripherals. As these products continue to become more sophisticated

and flexible, their technologies converge and their capabilities expand. In the world of consumer electronics, the whole has indeed become more than the sum of its parts.

The development of the microcomputer has accelerated an already rapid evolution, says David Allen, president of Boston Media Consultants and a writer specializing in TV production, computers, videodisks, and videotape. "They come along with greater speed. That's not a function of any interactivity, that's just a curve that the computer industry and microelectronics industry are on.

"Each development feeds the next development in a serendipitous way that makes succeeding developments faster to accomplish," says Allen. "You can really say that we're now to the point at which you could almost create any technological package you could conceive of, if you don't put a price restriction on it. Nothing is technologically impossible, in a broad sense. But it has to be accompanied by some kind of way to get return on investment. And that's what slows things down more than anything else right now. It's market-driven, not technologically driven."

During the past year, a parade of new technologies has entered the computer scene. The arrival of MIDI (Musical Instrument Digital Interface) has opened the doors to a new world of computer-based music composition and performance (see "Making Music with MIDI," *COMPUTE!*, January 1986). Laser-driven compact disc technology has branched out from stereo systems to computer data storage and retrieval. Smaller, less expensive video cameras and camcorders that connect with VCRs and computers are making inroads in consumer markets.

In addition, a new family of audio/video hardware and software products has been created to take advantage of the latest computers, particularly the Commodore Amiga, Atari ST, and Apple Macintosh.

It's appropriate that in this age of video one of the most promising fields of development is computer control of video images that originate from video cameras,

VCRs, laser disc players, other computers, or TVs with video outputs—essentially any device that puts out a composite video signal. For instance, Commodore is releasing two fascinating video peripherals for the Amiga: the Genlock, which plugs into the back of the Amiga and mixes external video signals with the computer's own video output; and the Amiga LIVE digitizer (formerly known as the "frame grabber"), which captures and digitizes an external video image in the Amiga itself.



Commodore/Amiga's Genlock accessory tucks beneath the rear of the Amiga computer and permits sophisticated video image mixing.

"Genlock is external to the Amiga and externally mixes two video sources, one of them the Amiga's," explains Paul Higginbottom, an Amiga product manager at Commodore. "So you take the Amiga's video source and the external video source, and you combine them—and the audio as well. Nothing comes into the Amiga with Genlock. With Amiga LIVE, a digitized picture is brought into the Amiga. So one [Genlock] is doing superimposing, and the other [Amiga LIVE] is actually taking an image and bringing it in.

"They operate separately, but you could certainly use them together," says Higginbottom. "You may want to take a real image and put Amiga's graphics on it, and digitize those back into the Amiga again."

Immediate applications for the Genlock include on-screen titling for video presentations or home movies, "electronic chalkboard" effects similar to those used for TV sports analysis, and special video effects achieved by mixing Amiga

graphics with other video images. At the Amiga's official unveiling in New York last summer, artist Andy Warhol used a video camera, Genlock, and Amiga LIVE to digitize a picture of rock singer Deborah Harry, then used a mouse-controlled graphics program to "paint" the video image with new colors. Amiga LIVE can be used not only for special video effects such as these, but also for video databases, says Higginbottom.

"We don't just mean pretty pictures. If you're a real estate agent or an architect, or you have a parts list you want to inventory, something like that—then you can have a video inventory," he explains. "And Amiga LIVE performs in real-time, not like most digitizers you see that usually take anywhere from 8 to 30 seconds to generate the picture on the screen. This is in real-time; if you have a movie camera, you'll see the image move as you move the camera."

Both the Genlock and Amiga LIVE are expected to be available in April or May, pending final FCC approval. Each accessory will cost about \$249.95.

A different video digitizer is in the works for the Atari ST and should be available by the time you read this. Hippopotamus Software is introducing the Hippovision Video Digitizer this spring for the ST and plans to have a version available later for the Amiga. (No price announced yet.)

"Anything that produces video signals, you just plug into the [digitizer] box that's connected to the computer," says Clint Ballard, vice president of engineering for the Los Gatos, California firm. "You press a button when you get a picture you like, and there you have it. We'll also have image processing software with which you can change around the colors—do whatever you want with it. This really opens up the graphics world."

For the Macintosh, which has a two-year head start on the Amiga and ST, there are already several video digitizers and compatible graphics programs available. *MacVision* from Koala Technologies, *Micro-Imager* from Servidyne Systems, Inc., *ThunderScan* from Thunderware, Inc., and a few others

COMPUTE!'s IBM LIBRARY

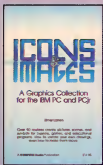
COMPUTE! offers you special savings on this set of top-selling titles for your IBM PC and PCjr.

Each book contains valuable tutorials, programming guides, personal and business applications, and games. Together, the books provide all the up-to-date, ready-to-use information and programs you need to get the most from your IBM personal computer.



COMPUTE!'s First Book of IBM

Edited, 326 pages
Thirty of the best games, utilities, graphics and sound generators, and applications for the IBM PC and PCjr. A disk is also available which includes programs in the book.
\$12.95, 010680SK.
ISBN 0-87455-010-8
\$14.95



Icons and Images: A Graphics Collection for the IBM PC and PCjr
Elmer Larsen, 227 pages
Ninety-four short routines to instantly enhance business, educational, and entertainment programs as either the IBM PC or PCjr.
ISBN 0-942386-84-1
\$14.95



Easy BASIC Programs for the IBM PC and PCjr
Brian Piyaa, 359 pages
Everything from games to home and office applications programs is included for the IBM PC and PCjr.
ISBN 0-942386-58-2
\$14.95



COMPUTE!'s Telecomputing on the IBM PC and PCjr
Arlan R. Levitas and Sheldon Leemos, 274 pages
The ins and outs of telecomputing on the IBM PC or PCjr, from selecting a modem and evaluating terminal software to getting online with the major information services.
ISBN 0-942386-98-5
\$14.95

ORDER ALL FOUR BOOKS FOR \$49.95 AND SAVE OVER 15% OFF THE RETAIL PRICE!

Take advantage of the great price savings and exceptional value of these bestselling books from COMPUTE! and order the four-book set today.

To order, call our toll free customer service number, 1-800-346-6767 (in NY 212-887-8525) and ask for COMPUTE!'s IBM Library. Or, mail the attached coupon with your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

You can also order individual copies of any of the above books at the stated retail price. Sorry, no substitutions.
LIMITED TIME OFFER! You must order before May 20 to receive your 15-percent discount.

Please send me the books I have selected. My payment is enclosed.

- _____ Sets of COMPUTE!'s IBM Library (four books per set) \$49.95
_____ COMPUTE!'s First Book of IBM (010-6) \$14.95
_____ COMPUTE!'s Telecomputing on the IBM (86-5) \$14.95
_____ Easy BASIC Programs for the IBM PC and PCjr (58-2) \$14.95
_____ Icons and Images: A Graphics Collection for the IBM PC and PCjr (84-1) \$14.95

- All orders must be prepaid
☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express
Subtotal _____
NY residents add 4.5% sales tax _____
Shipping and handling (\$2.00 per book, \$5.00 per book airmail) _____
Total amount enclosed _____

Account No. _____ Exp. Date _____ (required)
Name _____
Address _____
City _____ State _____ Zip _____
Please allow 4-6 weeks for delivery.

36417311

COMPUTE! Publications, Inc. 

One of the ABC Publishing Companies
525 7th Avenue, 9th Floor, New York, NY 10019

Subsidiaries of COMPTON: COMPTON U.S.A., Inc. (COMPTON) is a Group of COMPTON Books and COMPUTE!'s Apple Applications.

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anna's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from Holt, Rinehart, & Winston, 55 Horner Avenue, Toronto, ON M8Z 4X6.



THE ST™ COMPUTER LINE FROM ATARI®

IT'S LIKE GETTING THE POWER AND SPEED OF A FERRARI® FOR THE PRICE OF A FORD.®

When Atari introduced the 520ST™, we set the personal computer industry on its ear.

Nobody had ever produced a machine so powerful and technically advanced for such an incredibly low price. Nobody but Atari has done it yet.

The competition was stunned.

The critics wrote rave reviews. And consumers were ecstatic.

We could have rested on our laurels, but we didn't.

Instead, Atari extended the ST concept to a new computer called the 1040ST™.

The amazing new 1040ST is even more powerful than the 520ST and years ahead of all the competition at almost any price. The only question in

1986 isn't which company to buy a computer from, but which computer to buy from Atari.

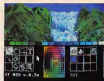
At \$799, the 520ST gives you 512 Kbytes of RAM, a high-resolution monochrome monitor, 2-button mouse, and 3.5" disk drive.

At \$999, the 1040ST gives you 1024 Kbytes of RAM, an ultra high-resolution monochrome monitor, 2-button mouse, and a built-in double-sided 3.5" disk drive, plus built-in power supply. Both the 520ST and the 1040ST can be connected directly to your own color TV. Or you can add an Atari RGB color monitor to get the sharpest, most colorful images possible. Add \$200 for color monitor.

It's simply a matter of choosing which model best fits your needs.

And whether you choose the 520ST or the 1040ST, you'll be getting the power and speed of a Ferrari for the price of a Ford.

In fact, you'll save hundreds and in some cases thousands of dollars over comparable computers. Which is why consumers are still ecstatic. Why the critics are still writing rave reviews. And why the competition is still stunned.



	ATARI® 1040ST™	COMPARABLES IN AMERICA™	IBM® PC/XT™	APPLE® Macintosh™	APPLE II®
Price	\$699	\$1795	\$975	\$1595	\$1295
CPU	68000	68000	80286	68000	6502C
Speed MHz	6.0	7.16	6.0	7.83	1.0
Standard RAM	1 MB	256K	256K	512K	128K
Standard ROM	161K	152K	64K	64K	19K
Number of Keys	85	80	85	60	63
Mouse	Yes	Yes	No	Yes	Optional
Screen Resolution (Non-Interlaced Mode)					
Color	640x200	640x200**	640x200	None	560x192
Monochrome	640x480	640x200***	720x350**	512x342	560x192
Color Output	Yes	Yes	Optional	None	Yes
Number of Colors	512	4096	16	None	16
Disk Drive	3.5"	3.5"	5.25"	3.5"	5.25"
Built-in Hard Disk (DMA) Port	Yes	No	Yes	No	No
Mod Interface	Yes	No	No	No	No
# of Sound Voices	3	4	1	4	1

Atari 1040ST with 128K RAM, \$749.

*Connects to standard color TV. For RGB color monitor add \$200.

**With optional Macintosh Keyboard (not 512-keypad).

***Interleave Mode - 640x480

Ferrari is a registered trademark of Ferrari SpA and Ferrari of America, Inc. Ford is a registered trademark of Ford Motor Company. IBM and PC/XT are registered trademarks of International Business Machines Corp. Commodore and Apple are trademarks of Commodore International U.S. Apple II, IIe and Macintosh are trademarks of Apple Computer, Inc. Atari 520ST, 1040ST and ST are trademarks of Atari Corp.

ATARI®

POWER WITHOUT THE PRICE.®

make excellent use of the Mac's high-resolution monochrome graphics. Since the Amiga and the ST each boast superb color graphics as well as high-resolution modes surpassing the Mac's, video digitization hardware and graphics software are becoming even more flexible and powerful.

As computers grow more capable of handling video images, other manufacturers are gearing up to take advantage of new markets expected to develop. Toshiba and Polaroid have announced products which strengthen the connections among computers, photography, and video. The two companies are jointly introducing a new instant video film recorder that produces instant color prints or slides from a TV set or monitor and has optional RGB (red-green-blue) computer input. The recorder features digital freeze-field capture, color preview capability, and accepts standard NTSC (National Television Standards Committee) signals.

The recorder captures and digitizes any image from a TV screen, whether the signal originated from a broadcast station, VCR, video camera, or any other standard video device. When equipped with the appropriate camera, the result is an instant photo print or 35 mm slide. With the push of a button, you could freeze one frame of your home movies, your favorite rock video, or a TV show, and then instantly produce a color picture. The recorder is expected to be available by midyear.

Polaroid is also introducing this year an improved version of its Palette computer image recorder. The Palette provides presentation-quality photos from computer graphics generated by a wide variety of computers, such as the Apple II series and the IBM PC family. It's capable of handling image resolutions up to 920 × 700, depending on the combination of hardware and software. Almost all presentation-graphics and graphics-editing software is compatible with the under-\$2,000 system.

Although few personal computer owners will spend several thousand dollars to buy such video systems for the home, the next few years will see

dramatic price drops as technology improves and costs decline.

For example, Kodak's Consumer Electronics Division plans to introduce a still video system that allows you to select and record individual video images. The system's player/recorder captures images in real time from any NTSC video signal and stores up to 50 images on a tiny floppy disk. An adjunct to this system is a film-to-disk transfer station that may be installed at film processors; you could have 35 mm color negatives transferred to the floppy disk, then view the pictures at home on your TV—ordering regular prints later, if you like.

Kodak had also planned to announce a new color video imager for producing instant prints of any video image. However, a recent decision by the U.S. Supreme Court on behalf of Polaroid has forced Kodak to withdraw from the instant photography business. Although Kodak had expected initial sales of the video imager to be in commercial and industrial applications, the long-range plan was to make the product part of home computer and video centers, according to Richard D. Lorbach, vice president of Kodak's consumer division.

"We anticipate that the color video imager eventually will be used as a home entertainment center component," said Lorbach before the court decision was handed down. "Our market research indicates that there is significant consumer interest in being able to make photographs of personal images displayed on TV screens."

This type of video system presents a wide range of possibilities. For example, by capturing images from your home videos, you could make a slide show of still shots or produce prints or slides for family albums. Computer artists could take their digital paintings or images captured from a video source and create their own sequenced video show. With the appropriate computer software, text could be overlaid on any of the images.

There are hundreds of business and industrial applications for this technology. Rather than spending thousands of dollars on outside production of sales and marketing presentations, almost any business

would have access to high-quality video production. A real estate agency could take photos or videotapes of its properties, add textual information on prices and other details, and then show the resulting package to their customers. Any of the frames could be turned into glossy prints for the house-hunters to keep for reference. The ramifications are virtually limitless.

One of the most important developments in the marriage of computer and video technology is the introduction of digital TVs—TV sets that convert the incoming analog broadcast signal into digital form. Toshiba, Sony, and most of the other large consumer electronics companies have invested millions of dollars to develop digital TV. Exceptionally clear pictures are only one of the benefits of this research. Digital TVs also have what's called PIP (picture-in-picture) capability—they can partition the viewing screen by opening separate "windows" for simultaneously displaying other video signals.

An example is the 26-inch DT-2680A TV receiver/monitor from NEC Home Electronics. It can simultaneously display the picture from the station that's tuned in plus moving pictures from any of three auxiliary video inputs, or color computer graphics through the set's RGB input. You can watch two channels at once, or a channel and a videotape, or even work with your home computer while watching TV on the same screen.

The picture you'll be watching is much sharper, too. Today's conventional TVs offer approximately 250 lines of horizontal screen resolution, while the NEC digital TV is capable of resolving up to 500 lines. This is actually more resolution than is available from broadcast signals. Through special filtering, the digital TV displays a broadcast screen resolution of 336 lines—the best that's possible with today's broadcasts.

In addition, the NEC digital TV has enough microprocessor-based memory to store up to three different still video pictures at a time. By pressing a button on the remote control, you can capture any video image and display it as an 8½-inch



The Computer Book Club®

Recognized as the authoritative source for micro users!

Superior Performance, Practical Price!

More programs, projects, ways to use your micro for home, hobby, education, and business!

Select 5 Books for Only \$2⁹⁵

Values to
\$123.75



1907
List \$23.95



1673
List \$21.95



1160
List \$14.95



1980
List \$19.95



1228P
List \$6.25 (paper)



1907
List \$21.95



1811
List \$21.95



1750P
List \$12.95 (paper)



1737
List \$18.95



1879
List \$23.95



1540
List \$22.95



1718P
List \$16.50 (paper)



1951
List \$19.95



1831P
List \$9.95 (paper)



1457
List \$17.95



1878P
List \$17.95 (paper)



1993
List \$21.95



1845
List \$18.95



1285P
List \$16.25 (paper)



1980
List \$16.95



1427P
List \$12.95 (paper)



1275
List \$15.95

Free guide to BASIC Statements & Commands



Plus FREE For Joining

7 very good reasons to join The Computer Book Club®

- **Big Savings.** Save 20% to 75% on books sure to increase your computer know-how
- **No-Risk Guarantee.** All books returnable within 10 days without obligation
- **Club News Bulletins.** All about current selections—mains, alternates, extras—plus bonus offers. Comes 13 times a year with hundreds of up-to-the-minute titles to pick from
- **Automatic Order.** Do nothing, and the Main selection will be shipped automatically! But... if you want an Alternate selection—or no books at all—we'll follow the instructions you give on the reply form provided with every News Bulletin
- **Bonus Books.** Immediately get Dividend Certificates with every book purchased and qualify for big discounts of 60% to 80%
- **Extra Bonuses.** Take advantage of added-value promotions, plus special discounts
- **Exceptional Quality.** All books are first-rate publisher's editions selected by our Editorial Board and filled with useful up-to-the-minute information

The Computer Book Club®

P.O. Box 80, Blue Ridge Summit, PA 17214

Please accept my membership in The Computer Book Club® and send the 5 volumes circled below, plus my FREE copy of BASIC Statements, Commands and Functions billing me only \$2.95 plus shipping and handling charges. If not satisfied, I may return the books within ten days without obligation and have my membership canceled. I agree to purchase 4 or more books at regular Club prices (plus shipping/handling) during the next 12 months, and may resign any time thereafter.

1000P 1160 1228P 1275 1295P 1407 1427P 1522 1540
1718P 1737 1748P 1750P 1811 1831P 1873 1878 1882 1884
1907 1945 1951 1960 1978P 1993 1997 2627 2648

Name _____
Address _____
City _____
State _____ Zip _____ Phone _____

Valid for new members only. Foreign applicants will receive ordering instructions. Canada must remit in U.S. currency. This order subject to acceptance by The Computer Book Club®.

CMPT-456

(diagonal) window within the 26-inch screen. Meanwhile, the background video image is unaffected. You could freeze-frame a fullback plying through the line while watching the play continue on the main screen.

As might be expected, the connection capabilities and special features of such a TV set go far beyond the few video and audio plugs found on even the better current sets. The NEC digital TV contains a stereo amplifier and stereo speakers, three sets of line video inputs for VCRs, video disc players, color cameras, and home computers, and an eight-pin RGB input. Outputs include a monitor jack that carries whatever is on the screen, a TV output that carries whatever channel is tuned, external speaker outputs, fixed audio line outputs for recording, and variable audio line outputs for volume-controlled connections to an external sound system.

As NEC vice president Gerry Tangney says, this "is a taste of the future of home TV." The NEC digital set is expected to be introduced in May, with the price to be announced soon.

Another new technology already on the horizon is high-definition TV (HDTV), an enhanced broadcast signal that offers 1,125 scan lines of information instead of the 525 now used in conventional American TV broadcasting. This would require broadcasters to upgrade their equipment, however, and efforts to adopt an HDTV standard have reportedly been riddled in international and corporate disagreements over how to bring about this doubling of screen clarity.

The growing popularity of compact disc (CD) audio players has given new impetus to the development and widespread consumer distribution of their digital data cousins, called CD-ROMs (Compact Disc-Read Only Memories). Although these laser discs are only 4.72 inches in diameter, they are capable of storing 600 megabytes of information on a single side, with an access time of seconds.

The first company out the door with CD-ROM players in the retail

market is the Subsystems and Peripherals Division of North American Philips Corporation. Its CM 100 disc player and CM 155 controller card works with the IBM PC-compatible computers (other interfaces will be announced this year). Available with the Philips CD-ROM player is Grolier's *The Electronic Encyclopedia*, the equivalent of a 20-volume reference collection on just about a quarter of one side of a CD-ROM disc. Although the initial purchase price of \$1,495 may keep initial sales out of the home market in volume, the price for CD-ROM technology is expected to drop quickly over the next couple of years.



Philips has introduced its CD-ROM drive which comes with Grolier's *Electronic Encyclopedia* on a compact disc. The entire package sells for \$1,495.

Technology occasionally moves in mysterious ways, and an example can be seen in new products which have taken advantage of the popularity—and intimidation—of word processors. Casio's new CW-30 Personal Typewriter blends the comforting familiarity of a typewriter with the ease of use of a computer word processor. The \$399.95 hybrid machine looks very much like a standard electric typewriter. But a quick look at the key-



This Casio computer-compatible electronic typewriter is a hybrid—part typewriter and part word processor—that can connect to a computer to serve as a printer.

board also shows a set of cursor and special function keys, plus a 15-character liquid-crystal display window for editing.

One of the most interesting features of the Casio typewriter is that it's computer-compatible. It contains both a Centronics-standard parallel interface and an RS-232 serial interface that lets the typewriter become a computer printer (plain or thermal paper). It can be hooked up to a 300 baud modem for uploading and downloading text with a computer. It has built-in pica and elite pitches, right justification, and multiple type fonts: boldface, underlining, double-wide characters, special symbols, and foreign alphabet characters. It has enough memory to store two pages of text, and with an optional memory expander, up to ten pages of text. Small removable memory cards let you save and store text. Casio obviously hopes to capture the best of both worlds, typewriters and word processors, at the same time it is attracting those who don't want to give up typewriters, but are fearful they're being left behind by word processors.



The Magnavox VideoWriter is an \$800 dedicated word processor aimed at the home market.

Magnavox has taken a different approach with its new VideoWriter, a dedicated home word processor that contains its own software, printer, spelling checker, and 18-line monitor (smaller than a regular computer screen, but larger than most portable computers). The \$800 VideoWriter has a memory capacity of approximately 70 pages of text, automatically stored on standard 3½-inch disks. While dedicated word processors have been used in offices for years, it's unusual to

see such a product for the home market, especially considering the number of people who buy multipurpose computers primarily for word processing.

Computers are converging with yet another technology, too—telephones. For example, Commodore is planning to introduce its new 1100 AnswerMate, a programmable computer-controlled telephone answering machine for the Amiga. The AnswerMate connects to the Amiga's RS-232 port and to a telephone. Not only does it play back your taped greetings and record messages, but it also can respond with messages generated by the Amiga's built-in synthesized voice. And multitasking software included with the AnswerMate lets it answer phone calls while you're busy using the computer for other things. (Price to be announced.)




Commodore's AnswerMate connects to the Amiga computer to serve as a telephone answering machine that can make use of the Amiga's multiprocessing and synthesized speech capability.

There is scarcely an area of consumer electronics which is not moving either directly or indirectly toward the personal computer, either as a peripheral or as a microprocessor-based stand-alone device. Even the ways in which computer users receive their software may be undergoing change in the future.

For example, Cauzin Systems, with backing from Kodak, has developed the Softstrip system of information storage. Data is encoded on a strip of paper in a format similar to—but more compact than—the familiar bar codes found

on consumer products. One strip, which typically measures $\frac{9}{16}$ by $\frac{3}{8}$ inches, can store up to 5,500 characters (about three typewritten pages). The strips can be printed on ordinary paper and are read by an electro-optical scanner. Connected to a computer, the scanner reads the coded strips and transfers the data into memory for later storage on disk.

Further examples of converging electronics technologies abound in virtually every field. The emergence of stereo TVs and VCRs, coupled with a stereo-capable computer such as the Amiga, obviously opens new possibilities for audiophiles. Interactive video, spurred by improvements in laser discs, is another rapidly evolving technology with a connection to personal computing. Radio signals relayed by satellites can carry data accessible by computer users. Use of electronic mail systems is expected to jump from less than a billion messages a year today to more than 20 billion by the end of the decade, ultimately becoming a major service as common as the telephone and the U.S. mails.

As media consultant David Allen noted earlier, technology is capable of virtually anything today; but the successful marketing of an idea is the key to its success. In the foreseeable future, neither technology nor the marketplace shows any signs of slowing down. 

Attention Programmers

COMPUTE! magazine is currently looking for quality articles on Commodore, Atari, Apple, and IBM computers (including the Commodore Amiga and Atari ST). If you have an interesting home application, educational program, programming utility, or game, submit it to COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Or write for a copy of our "Writer's Guidelines."

[illegible]

Report From:

The Winter Consumer Electronics Show

A Turning Point For Atari?

Tom R. Halfhill, Editor

Following up its strong showing at the Comdex computer show in November, Atari introduced a more powerful version of its ST at the Winter Consumer Electronics Show in January. Thanks to increasing sales, growing software support, widening distribution, and hints of new enhancements to come, industry watchers are suddenly taking more notice of Atari's bid for a comeback. Meanwhile, Commodore also entered 1986 with encouraging sales and Apple is responding with an improved Macintosh and lower prices.



A year ago it seemed impossible. Commodore founder Jack Tramiel had split with his successful computer company after a management dispute, bought the debt-ridden Atari that he had nearly destroyed in price wars, installed his sons in key positions, laid off most of the work force, rushed the design of a powerful 16/32-bit machine in only six months, introduced it at an unheard-of low price, and announced he was going to resurrect Atari as a major contender in the personal computer marketplace.

Atari still isn't home free. But the house that Jack built suddenly seems a lot more solid.

Strengthened by encouraging sales of the 520ST—according to estimates, at least 100,000 units worldwide through Christmas—Atari is now attracting more attention within the industry. "Frankly, a lot of people didn't think Jack would make it this far," says one observer. "Now they're

taking him a lot more seriously."

If Atari's comeback ultimately succeeds, the six-week period between late November 1985 and early January 1986 may well become recognized as the turning point. During that period, Atari piled up sizeable holiday sales and made impressive appearances at two crucial industry trade shows: Fall Comdex and the Winter Consumer Electronics Show. Both are held annually in Las Vegas and are among the largest trade shows in the U.S., with upwards of 100,000 people at each event.

At Comdex, which is oriented toward business computing, Atari demonstrated to skeptics that its 520ST was a real machine with dozens of software packages. At CES, a show that encompasses every consumer electronics product imaginable, Atari was the only major computer manufacturer in attendance and made three important announcements: the new 1040ST, a more powerful version of the 520ST with one megabyte of memory and a built-in disk drive; price reductions of \$100 for the monochrome and color 520ST systems; and a shift to mass-market outlets such as department stores for the 520ST.

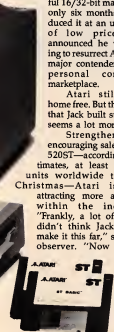
Atari's appearances at Comdex and CES seemed all the more impressive due to the conspicuous absence of its closest competitor, Commodore. People were surprised when Commodore missed Comdex because the company has been trying to position the Amiga as a business computer and Comdex was the ideal place. But there was shock when Commodore bowed out of CES because Commodore has never missed a CES since the days when wristwatches and calculators were its stock in trade.

Commodore didn't have much to say about missing the shows. However, one Commodore executive admitted he was "uneasy" about the reaction at CES—ironically, the rumors of imminent financial catastrophe that once followed Atari were now being whispered about Commodore. The rumors proved untrue, however, and Commodore says it definitely plans to attend the Summer CES in Chicago this June.

Actually, Commodore finished 1985 with heavy sales of its own. According to reliable estimates, Commodore sold about one million 64s, as many as 500,000 Commodore 128s, and at least 20,000 Amigas. Even Commodore was caught off guard by the 64 and 128 sales. In fact, insiders say Commodore tried twice during the fall to discontinue the 64, but had to restart production both times to meet sudden demand. As an indication that Amiga sales are healthy, the leading independent software supplier for the computer—Electronic Arts—says it recovered all of its 1985 Amiga development costs within two weeks after releasing its first Amiga products.

Although Atari and Commodore are still struggling financially, both have survived a rough market in 1985 and appear to be in better shape for 1986.

Since Atari was the only major computer company exhibiting at CES (IBM and Apple routinely avoid this show), most of the computer news was Atari-related. The main event was the introduction of the 1040ST, the first one-megabyte computer selling for under \$2,500. In fact, it's the first one-megabyte computer selling for under \$1,000.



The new Atari 1040ST is the first one-megabyte computer for under \$1,000. It has 1,024K of RAM and a built-in, double-sided disk drive.

The 1040ST is basically an enhanced 520ST and is fully compatible with existing ST software and hardware. The keyboard and all interfaces are identical: RS-232 serial and Centronics-standard parallel ports; in/out MIDI (Musical Instrument Digital Interface); floppy and hard disk interfaces; plus monochrome and analog RGB monitor outputs. The graphics-oriented user interface, GEM (Graphics Environment Manager), is the same.

New features on the 1040ST include one megabyte (1,024K) of Random Access Memory (RAM), twice the amount that comes with the 520ST; a built-in, double-sided 3½-inch disk drive with a capacity of 770K per disk, twice the capacity of the drive sold with the 520ST; a TV output jack; and an internal power supply for both the computer and drive, reducing the familiar clutter of external cables.

Like recent-model 520STs, the 1040ST comes with its Tramiel Operating System (TOS) in Read Only Memory (ROM) chips, freeing up more than 200K RAM that used to be required when loading TOS from disk. Also like the 520ST, the 1040ST comes with ST BASIC, the NEOchrome graphics-drawing program, and a word processor, 1st Word (the 520ST comes with ST Writer). Atari says 1st Word has GEM features such as drop-down menus and on-screen type fonts. (ST Writer, by contrast, is a direct translation from the AtariWriter word processor for eight-bit Ataris.)

There are two different 1040ST packages. With a high-resolution monochrome monitor, the suggested retail price is \$999.95. With an analog RGB color monitor, the price is \$1,199.95. Atari says the 1040ST will be sold only through computer dealers and should be available immediately.

The 520ST also underwent some minor changes. The latest models will be shipped with TOS in ROM and a TV output jack. To widen distribution, the 520ST will be sold through mass-market outlets in three different ways. A system that includes the computer, a single-sided 3½-inch disk drive (380K capacity), and hi-res monochrome monitor will now

be priced at \$699 suggested retail. The same system with an analog RGB color monitor instead of the monochrome screen will be priced at \$899. Both prices are \$100 lower than before. The 520ST components are also available separately: \$399 for the computer, \$199 for a single-sided drive, \$299 for a double-sided drive, \$199 for the monochrome monitor, and \$299 for the RGB monitor.

by lining up a series of computers running the now-famous Amiga bouncing ball demo. (A screen photo of this demo appears in COMPUTE!'s cover story on the Amiga in September 1985; it shows a red-and-white checkered globe spinning and bouncing around the screen, casting a transparent shadow on the background.) An Amiga, 520ST, Macintosh, and eight-bit Atari 130XE spent four straight



Atari's 520ST has been improved with a ROM-based operating system, a TV output jack, and a \$100 lower price. Also, for the first time, the computer and other components will be available separately in mass-market outlets.

Rumors abounded at CES about new developments for the ST line, including a better graphics chip, a bit-block transfer chip similar to the one in the Amiga, a 5¼-inch disk drive adapter for use with an IBM PC emulator, and more. Officially, Atari won't confirm or deny if it's preparing to introduce any of these products in the near future.

However, Atari is expected to announce at least one enhancement at an upcoming computer show in West Germany (about half of all ST sales are in Europe). The most likely possibility is the bit-block chip, which allows faster screen graphics. Also, it is now known that three companies outside Atari are working on PC emulators for the ST, each taking a slightly different approach. At this writing (mid-January), none of them is expected to be ready for several months.

Atari's CES exhibit poked fun at the Amiga and Apple Macintosh

days dribbling checkered balls at the Atari booth. Oddly enough, the 130XE version was perhaps the most impressive of all. Not only was the 130XE bouncing a checkered ball, but also a 3-D image of the Atari logo decorated with 128 rippling colors.

Apple hasn't been oblivious to the competition, and a week after CES announced an improved version of the Macintosh and lower prices for the 512K Mac. The new Macintosh Plus has one megabyte of RAM, a double-sided disk drive that stores 800K (twice the capacity of existing Macintosh drives), a redesigned keyboard with numeric keypad and cursor keys, a faster operating system, and an extra peripheral port called the Small Computer System Interface. The suggested retail price is \$2,599. The 512K Mac was reduced from \$2,499 to \$1,999, and kits are available so owners of 128K and 512K Macs can upgrade to the Mac Plus.

FREE SOFTWARE

Now Get Up To 200 FREE Programs* When You Subscribe to **COMPUTE!** Today.



Subscribe to **COMPUTE!** today and you'll be getting a lot more than just another computer magazine. That's because **COMPUTE!** comes complete with up to 20 FREE programs in each big issue.

Subscribe now and you can depend on a steady supply of high quality, fun-filled programs like Cash Flow Manager, Speed Ski, Turtle Pilot, Boggler, Text Plot, Retirement Planner, and hundreds of other educational, home finance, and game programs the entire family can use all year long.



The free programs alone are worth much more than the low subscription price. But there's more to **COMPUTE!** than just free programs.

COMPUTE!'s superb articles deliver the latest inside word on everything from languages to interfaces...programming to disk drives. And our up-to-the-minute software reviews are must reading for any home user.



Whether you're a novice or an experienced user, **COMPUTE!** is perfect for you. So subscribe today. Return the enclosed card or call 1-800-247-5470 (in Iowa 1-800-532-1272).

COMPUTE! Publications, Inc. 

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of: COMPUTE! COMPUTE! Source COMPUTE! Source Disk COMPUTE! Source and COMPUTE! Source Applications

*Ranging from short techniques and programming tips to complete, powerful applications programs.

Atari didn't ignore its older products at CES. Among other things, two new eight-bit computer packages were unveiled.

The 130XE, a 400/800/XL-compatible computer with 128K RAM that was introduced last year, will now be available in a \$399 package that includes a 1050 disk drive, 1027 printer, and five pieces of software: *AtariWriter*, *Music Painter*, *Paint*, *Star Raiders*, and *Silent Butler*. The 130XE is still available separately for \$149.

Atari's new lowest-end computer is the 65XE, a 400/800/XL-compatible machine with 64K

The revamped 2600 is smaller, lighter, sleeker, has a carrying handle, and costs less than \$50. The 7800 machine was one of the last projects of the old Atari before Tramiel took over, and is now being unpacked from mothballs. Thanks to a chip named MARIA, the 7800's graphics are superior even to those of the eight-bit computers, and the machine accepts all 2600 cartridges without an adapter. It sells for less than \$80 and comes with a new version of *Pole Position*, the hit car racing game.

Computers aren't much good without software, and Winter CES demonstrated that a lot of companies are bringing out new programs for the ST and Amiga as well as popular eight-bit machines. Unfortunately, there were a lot fewer software companies at this CES than last year's. The West Hall, a large building which supplements the main Convention Center, once was filled with software publishers. This year, only a handful of them shared space with companies selling satellite dishes, videotapes, cable TV accessories, and other peripherals. The last two years have been tough, and many software developers either can't afford to exhibit at CES anymore or are out of business altogether.

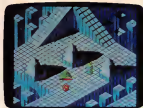
Still, some fascinating products are on the way. Starting with ST and Amiga software, here's what's new:

RAM. It was actually announced at last year's Winter CES, but was withheld from the market until existing inventories of 800s and 800XLs were sold out. The 65XE will be sold separately for under \$100 or in a package similar to the 130XE's for \$300-\$350.

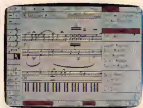
Much to everyone's surprise, Atari also came to CES with a redesigned 2600 videogame machine (formerly known as the VCS) and the new 7800 game machine. What's that, you thought videogame machines were dead? Guess again. Atari says it sold over a million 2600s in 1985. Apparently many of them are going to new markets overseas—including, we hear, the People's Republic of China.



Arcticfox is a new strategic tank game for the Amiga from Electronic Arts.



This is the Amiga version of *Marble Madness*, but Electronic Arts is also bringing out an Atari ST version of the game.



Deluxe Music is a note-oriented composition program designed for the Amiga by Electronic Arts.



Even if you don't know much about music, you can play songs on an Amiga with Electronic Arts' *Instant Music* program. It has numerous built-in instrument sounds and doesn't rely on standard musical notation.

GET THE KNOW-HOW TO REPAIR EVERY COMPUTER ON THIS PAGE.

Learn the Basics the NRI Way—and Earn Good Money Troubleshooting Any Brand of Computer

The biggest growth in jobs between now and 1995, according to Department of Labor estimates, will occur in the computer service and repair business, where demand for trained technicians will actually double.

You can cash in on this opportunity—either as a fulltime corporate technician or an independent service-person—once you've learned all the basics of computers the NRI way. NRI's practical combination of "reason-why" theory and "hands-on" building skills starts you with the fundamentals of electronics, then guides you through advanced electronic circuitry and on into computer electronics. You also learn to program in BASIC and machine language, the essential languages for troubleshooting and repair.

You Build—and Keep—a 16-bit Sanyo personal computer

The vital core of your training is the step-by-step building of the 16-bit Sanyo MBC-550 series computer. Once you've mastered the details of this state-of-the-art machine, you'll be qualified to service and repair virtually every major brand of computer, plus many popular peripheral and accessory devices.

With NRI training, you learn at your own convenience, in your own home. You set the pace—without classroom pressures, rigid night-school schedules, or wasted time. You build the Sanyo IBM compatible computer from the keyboard up, with your own personal NRI instructor and the complete NRI technical staff ready to answer your questions or give you guidance and special help whenever you need it.

Praised by critics as the "most intriguing" of all the IBM-PC compatible computers, the new Sanyo uses the same 8088 microprocessor as the IBM-PC and



IBM is a Registered Trademark of International Business Machine Corporation.
Space is a Registered Trademark of Space America, Inc.
Apple and the Apple logo are Registered Trademarks of Apple Computer, Inc.
Compaq is a Registered Trademark of COMPAQ Computer Corporation.
©1984 AT&T Technologies, Inc.

the 8088 microprocessor functions, using machine language. You'll also prepare the interfaces for future peripherals such as printers and joysticks.

100-Page Free Catalog Tells More

Send the coupon today for NRI's big 100-page color catalog on NRI's electronics training, which gives you all the facts about NRI courses in Microcomputers, Robotics, Data Communications, TV/Video/Audio Servicing, and other growing high-tech career fields. If the coupon is missing, write to NRI Schools, 3939 Wisconsin Avenue, NW, Washington, DC 20016.

AND MORE.

features the MS/DOS operating system. As a result, you'll have a choice of thousands of off-the-shelf software programs to run on your completed Sanyo.

Your NRI course includes installation and troubleshooting of the "intelligent" keyboard, power supply, and disk drive, plus you'll check out

Your NRI course

includes the

Sanyo 16-bit

IBM compat-

ible com-

puter with

128K RAM,

monitor,

double density/

double sided disk drive,

and "Intelligent" keyboard; the NRI

Discovery Lab[®], teaching circuit design and operations;

a Digital Multimeter; Bundled Speed Sheet and Word

Processing Software, worth over \$1000 at retail—and more.



NRI SCHOOLS
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, Washington, DC 20016
We'll give you tomorrow.

☒ CHECK ONE FREE CATALOG ONLY

- ☐ Computer Electronics with Microcomputers
- ☐ Data Communications
- ☐ Robotics & Industrial Controls
- ☐ Video Electronics Servicing
- ☐ Electronic Design Technology
- ☐ Digital Electronics

- ☐ Scientific Communications
- ☐ Communications Electronics
- ☐ Industrial Electronics
- ☐ Basic Electronics
- ☐ Telephone Servicing
- ☐ Small Engine Servicing
- ☐ Appliance Servicing

- ☐ Automotive Servicing
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction
- ☐ Locksmithing & Electronic Security

For Career courses approved under GI Bill,
☐ check for details

Name (Please Print) _____

Age _____

Street _____

City/State/Zip _____

Accredited by the National Home Study Council

126-046

shipped 15,000 copies of *Deluxe-Paint* the first two weeks.) Scores composed with *Deluxe Music* can also be integrated with *Deluxe Video Construction Set*.

ST software from Electronic Arts includes two programs already released for the Amiga: *Financial Cookbook*, a home financial planner, and *Marble Madness*, an arcade game. Prices for Electronic Arts' Amiga and ST software range from \$39.95 to \$79.95.

Hippopotamus Software (Los Gatos, California) is bringing out the Hippo ST Sound Digitizer and Hippovision Video Digitizer for the Atari ST. The sound digitizer plugs into the printer port and lets you sample, modify, and play back any type of sound. It includes special-effects software and a microphone for \$139.95. The video digitizer lets you capture images in 256 X 256-pixel resolution from any composite video source, such as a video camera, videocassette recorder, videodisc player, or TV tuner. Picture files are compatible with Atari's NEOchrome drawing program, can be printed on color printers, and can be transmitted via modem. The price was not announced.

Hippopotamus has 13 other ST programs scheduled for release soon, including *HippoWord*, an \$89.95 word processor; *HippoConcept*, an idea processor, \$89.95; *HippoSimple*, a database manager, \$49.95; *Hippo Disk Utilities*, \$49.95; *HippoBackgammon*, \$39.95; *HippoSpell*, a spelling checker with 30,000-word dictionary and user-definable terms, \$39.95; *Hippo RAMdisk*, \$34.95; *Hippo Computer Almanac*, which contains over 35,000 facts on everything from area codes to sports trivia, \$34.95; *Hippo Jokes & Quotes*, with selectable PG, R, or X ratings, \$34.95; *HippoArt I*, a collection of 30 picture files compatible with NEOchrome, \$39.95; *Hippo EPROM Burner*, for programming your own chips, \$139.95; *HippoClean*, a disk drive cleaning kit, \$29.95; and *Hippo-Pixel*, a utility for creating your own sprites and fonts, \$39.95.

Aegis Development (Santa Monica, California) is bringing out four graphics products for the Amiga: *Aegis Images*, a drawing program; *Aegis Draw*, a Computer-Aided Design (CAD) program; *Ae-*

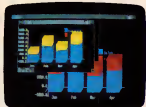


This strikingly beautiful picture was created on the Amiga's 320 X 200-pixel screen in 32 colors with *Aegis Images*, a graphics-art program demonstrated at CES.

gis Animator, for creating animated sequences; and *Impact*, an executive graphics package. *Images*, *Animator*, and *Impact* were actually developed by Island Graphics of Sausalito, California. Under an agreement with Commodore, they were supposed to be released under the Amiga brand name. For instance, *Images* was originally known as *Graphicraft* or *ProPaint* (several screens created with this program appear on the Amiga's packaging and in the September 1985 issue of *COMPUTE!*). But Island Graphics and Commodore had a falling out, and the Amiga *Graphicraft* currently being sold is not the software developed by Island

Graphics. Instead, Aegis acquired the marketing rights to the Island Graphics programs, enhanced them, and renamed them *Aegis Images*, *Animator*, and *Impact*.

Images is available for \$79.95 separately, or for \$139.95 in a package with *Aegis Animator*. Using *Images* screens as a backdrop, *Animator* allows 3-D animation and rotation, metamorphic shape manipulation, storyboarding of up to nine separate sequences, and ghost-line animation. *Impact* (\$199.95) is for business presentation graphics and includes a slide show feature—charts, graphs, and pictures can be flipped in a predetermined sequence and transformed.



Impact, another product from Aegis Development, makes it possible to produce three-dimensional business graphics using the Amiga's 4,096 colors.



Aegis Draw is a Computer-Aided Design (CAD) program for the Amiga which is aimed at professional users.

C-128[™] LIBRARY

and C-64[™]

COMMODORE 128[™] INTERNALS  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	COMMODORE 128[™] BASIC 7.0 INTERNALS  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	COMMODORE 128[™] TRICKS & TIPS BEST SELLER  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	COMMODORE 128[™] 1571 INTERNALS  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	COMMODORE 128[™] CAD 128[™] COMPUTER AIDED DESIGN  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>
--	---	---	--	---

Detailed guide presents the 128's operating system, explains graphics chips, memory management unit, 40 pin graphics and formatted ROM listings. **400pp \$19.95**

Get all the inside information on BASIC 7.0. This exhaustive handbook is complete with formatted BASIC 7.0 ROM listings. Coming Summer '88. **\$19.95**

Filled with info for everyone. Covers 30 columns, hi-res graphics, windowing, memory layout, Vram listings, sprites, software protocols, evaluating. **300pp \$19.95**

Insider's guide for novice & advanced users. Covers sequential & relative files, & direct address commands. Describes DOS routines. **400pp \$19.95**

Learn fundamentals of CAD while developing your own system. Design objects on your screen to dump to a printer. Includes listings for 54-word Simon's Basic. **300pp \$19.95**

COMMODORE 128[™] COMPLETE BASIC HANDBOOK  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	COMMODORE 128[™] PEEK & POKES  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	COMMODORE 128[™] CP/M ON THE C-128  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	THE UNOFFICIAL BOOK FOR THE COMMODORE 128 TRICKS & TIPS FOR THE COMMODORE 128  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>	ADVENTURE GAMEWRIter's HANDBOOK FOR COMMODORE 128  <small>A DATA RESEARCH BOOK PUBLISHED BY Abacus DATA Software</small>
--	--	---	---	---

Introduction to programming, problem analysis, thorough description of all BASIC commands with hundreds of examples, monitor commands, utilities, much more. **400pp \$19.95**

Presents dozens of programming quick-tips. Easy and useful techniques in the operating system, disks, zero page, pointers, the BASIC interpreter and more. **\$19.95**

Essential guide for everyone interested in CP/M on the 128. Simple explanation of the operating system, memory usage, CP/M utility programs, source files & more. **\$19.95**

Adventure Gamewriter's Handbook. Step-by-step guide to designing and writing your own adventure games. With automated adventure game generator. **200pp \$14.95**

Peek & Pokes for the C-64. Includes in-depth explanations of PEEK, POKE, LDR, and other BASIC commands. Learn the "inside" tricks to get the most out of your 64. **200pp \$14.95**

ANATOMY OF C-64 Insider's guide to the 64 internals. Graphics, sound, I/O, kernel, memory maps, more. Complete formatted ROM listings. **300pp \$19.95**

ANATOMY OF 1541 Drive Belt handbook on floppy controllers. Many examples and detailed, fully commented 1541 ROM listings. **500pp \$19.95**

MACHINE LANGUAGE C-64 Learn 6502 code with test programs. Many examples and listings for complex assembler, monitor, & simulator. **200pp \$14.95**

GRAPHICS BOOK C-64 - best reference covers basic and advanced graphics. Sprites, animation, hires, Multicolor, highpen, 3D-graphics, RGB, CAD, plotter, joysticks, curves, more. **350pp \$19.95**

TRICKS & TIPS FOR C-64 Collection of easy-to-use techniques. Advanced graphics, improved data input, enhanced BASIC, CP/M, more. **275pp \$19.95**

1541 REPAIR & MAINTENANCE Handbook describes the disk drive hardware, includes schematics and techniques to keep 1541 running. **200pp \$19.95**

ADVANCED MACHINE LANGUAGE Not covered elsewhere: video controller, interrupts, timers, clocks, I/O, real time, extended BASIC, more. **210pp \$14.95**

PRINTER BOOK C-64/VIC-20 Understand Commodore, Epson-compatible printers and 1520 plotter. Packaged utilities; graphics dump; 30-pin; commented MP5001 ROM listings, more. **330pp \$19.95**

SCIENCE/ENGINEERING ON C-64 In depth intro to computers in science. Topics: chemistry, physics, biology, astronomy, electronics, others. **350pp \$19.95**

CASSETTE BOOK C-64/VIC-20 Comprehensive guide; many sample programs. High speed operating system file loading and saving. **220pp \$14.95**

IDEAS FOR USE ON C-64 Thermostats, auto response, calculator, recipe file, stock lists, diet planner, window advertising, others. Includes listings. **200pp \$12.95**

COMPILER BOOK C-64/C-128 All you need to know about compilers: how they work; designing and writing your own; generating machine code. With working example compiler. **300pp \$19.95**

ADVENTURE GAMEWRIter's Handbook Step-by-step guide to designing and writing your own adventure games. With automated adventure game generator. **200pp \$14.95**

PEEK & POKES FOR THE C-64 Includes in-depth explanations of PEEK, POKE, LDR, and other BASIC commands. Learn the "inside" tricks to get the most out of your 64. **200pp \$14.95**

Optional Diskettes for books. For your convenience, all programs contained in each of our books are available on diskette to save you time entering them from your keyboard. Specify name of book when ordering. **\$14.95 each**



Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Call now for the name of your nearest dealer. Or to order directly by credit card, MC, AMEX or VISA call (616) 241-5510. Other software and books are available—Call and ask for your free catalog. Add \$4.00 for shipping per order. Foreign orders add \$10.00 per book. Dealer inquiries welcome—1200+ nationwide.

C-128 and C-64 are trademarks of Commodore Business Machines.

Aegis Draw, for hobby and professional CAD work, actually consists of two programs: *Aegis Draw* and *Aegis Draw Professional*. The first version retails for \$199.95 and can be upgraded to the *Professional* version at extra cost. *Aegis Draw* has zooming, automatic scaling, selectable grids, layering, and multiple windows so you can work on several drawings simultaneously, or on different parts of the same drawing. It supports Kurta and Summagraphics digitizers, and plotters by Roland, Hewlett Packard, Houston Instruments, Epson, and Comrex. All four Aegis programs for the Amiga should be available immediately.

Supra Corporation (Albany, Oregon) announced a series of hard disk drives for the Atari ST with capacities of 10, 20, 40, and 80 megabytes. Prices start at \$799 for 10 megabytes. They should be available this spring.

Optimized Systems Software (San Jose, California) was demonstrating *Personal Pascal* for the ST at the Atari booth and has already started shipping. This language supports all of the ST's special features, compiles and links most programs in about a minute (even faster with a hard disk or RAM disk), and sells for \$74.95. OSS is also readying a version of the Prolog language and a set of disk utilities for the ST.

Cardco (Wichita, Kansas), known in the past for its VIC-20 and Commodore 64 add-ons, is preparing a one-megabyte memory expansion board for the Amiga. The board plugs into the expansion bus, and Cardco says it will be available this spring for about \$400.

Unison World (Berkeley, California) is converting *PrintMaster*—a printer utility similar to *The Print Shop*—to the Atari ST. It's already available for the Commodore 64, IBM, and CP/M computers.

Activision (Mountain View, California) said sales of its initial software for the ST and Amiga have been "quite strong" and that additional titles will be released this year. These include Garry Kitchen's *GameMaker: The Computer Game Design Kit* and *The Music Studio* for the Amiga and ST; and *The Activision Little Computer People Discov-*



Another new music-composition program is *The Music Studio* from Activision. There are versions for the Amiga, Atari ST, and IBM PCjr/Tandy 1000 computers.

ery Kit for the Amiga. *GameMaker* lets you write videogames without learning a programming language. Finished games can be saved on a blank disk and run independently of the master program, so you can distribute copies to friends. Activision is sponsoring a *GameMaker* contest—first prize is a trip to Activision plus \$5,000.

The Music Studio is a composition tool designed by Audio Light for both amateur and professional musicians. With it, you can design instruments and create new sound effects. The Amiga version plays up to 16 simultaneous sound channels in stereo.

Accolade/FTL Games (Cupertino, California) has released *Sundog: Frozen Legacy* for the ST. Originally written for the Apple II series, *Sundog* is a graphics strategy game that has been completely redesigned to take advantage of the ST's advanced graphics. Thanks to a proprietary data-compression scheme, hundreds of different full-color screens are stored on the program disk.

Mindscape (Northbrook, Illinois) is introducing three programs



Mindscape's *Deja Vu: A Nightmare Comes True* is a 1940s-style mystery game for the Amiga and Macintosh.

for the Amiga and one for the ST. *Brataccas* (\$49.95) is a graphics adventure game for both computers that was developed by Psynosis Limited of England. It was written specifically to take advantage of the 68000 chip inside the Amiga and ST. In *Brataccas*, you're a scientist who has invented a genetic process for creating a superbeing. With an evil government and the underworld in pursuit, you flee to a colonized asteroid, Brataccas. The object is to expose the government's corruption and clear your name. Brataccas is populated with nearly 60 different characters.

For the Amiga only, Mindscape is releasing *The Halley Project: A Mission in Our Solar System* (\$49.95), a realtime simulation of the solar system developed by Tom Snyder Productions with help from the Massachusetts Institute of Technology; *Deja Vu: A Nightmare Comes True* (\$54.95), a 1940s-style mystery game; and *Keyboard Cadet* (\$39.95), a typing tutor. (Incidentally, Mindscape is the company which wrote the *Amiga Tutor* supplied with the Amiga.)

Abacus Software (Grand Rapids, Michigan) is importing a professional-quality program called *PC Board Design* for the ST. When Abacus finishes translating this circuit-designing utility from German, it will sell for \$395.

Batteries Included (Richmond Hill, Ontario) was demonstrating its D.E.G.A.S. drawing program for the ST with a slideshow of screens called up in rapid sequence from a hard disk. D.E.G.A.S. started shipping just before CES, and it's already a hit—Batteries Included says that sales figures for the first two weeks were greater than for any other program in its history. ST and Amiga versions of the *Isgwr Portfolio System*, a stock-management program, are scheduled for release later this year at \$249.95.

Q-R-S (Buffalo, New York), a company that started back in 1900 by making music rolls for player pianos, is releasing its digital music library for the Amiga and ST. This consists of a number of disks containing piano music by Joplin, Gershwin, Liberace, and other artists and composers. Each disk contains six songs and sells for \$19.95.

Sierra On-Line (Coarsegold,

'128™ and C-64™ SOFTWARE



Our BASIC Compilers are the complete compiler and development packages. Speed up your programs from \$5 to \$95.

Our BASIC Compilers give you many options: flexible memory management; choice of compiling to machine code, compact p-code or a mixture of both. Also on the '128, 40 or 80 column monitor output and FAST-mode operation.

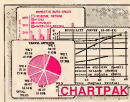
The '128 Compiler's extensive 80-page programmer's guide covers compiler directives and options, two levels of optimization, memory usage, input/output handling, 80 column hi-res graphics, faster, higher precision math functions, speed and space saving tips, more. A great package that no software library should be without.

BASIC 128 Compiler \$59.95
BASIC 64 Compiler \$39.95



For school or software development, select **SUPER C**. Learn to use one of today's most popular languages on your Commodore 128. Powerful screen editor, compiler, linker-link up to seven modules, I/O library-includes printf and fprintf. Compile your C programs into fast machine language.

C-128 \$79.95
C-64 \$79.95

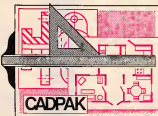


For the professional who wants to easily create high quality charts and graphs without programming. You can immediately change the scaling, labeling, axis, bar-filling, etc. to suit your needs. Accepts data from CalcResult and MultiPlan. C-128 version has 3X the resolution of the '64 version. Outputs to most printers.

C-128 \$39.95
C-64 \$39.95

PowerPlan

One of the most powerful spreadsheets with integrated graphics for your Commodore computer. Includes menu or keyword selections, online help screens, field protection, windowing, trig functions and more. Power-Graph, the graphics package, is included to create integrated graphs & charts. **C-64 \$39.95**



CADPAK is a remarkably easy to use drawing package for accurate graphic designs.

Using CADPAK's new dimensioning features you can create exact scaled output to all major dot-matrix printers.

This enhanced version of CADPAK allows you to input via the keyboard or a high quality lightpen. Two graphic screens permit you to COPY from one screen to another. DRAW, LINE, BOX, CIRCLE, ARC, ELLIPSE are but a few of the many selections to choose from. FILL objects with preselected PATTERNS; add TEXT; SAVE and RECALL designs to/from disk. You can define your own library of choice symbols/objects with the easy-to-use OBJECT MANAGEMENT SYSTEM-It will store up to 104 separate objects.

C-128 \$59.95
C-64 \$39.95



Compiler and Software Development System

Not just a compiler, but a complete system for developing applications in Pascal. Extensive editor with search, replace, auto, renumber, etc. Standard J & W compiler that generates fast machine code. If you want to learn Pascal or to develop software using the best tools available-SUPER Pascal is your first choice.

C-128 \$59.95
C-64 \$39.95

OTHER TITLES AVAILABLE:

Technical Analysis System

A sophisticated charting and technical analysis system for serious investors. By charting and analyzing the past history of a stock, TAS can help pinpoint trends & patterns and predict a stock's future. Enter data from the keyboard or from online financial services.

C-64 \$59.95

Personal Portfolio Manager

Complete portfolio management system for the individual or professional investor. Easily manage your portfolios, obtain up-to-the minute quotes and news, and perform selected analysis. Enter quotes manually or automatically through Warner Computer Systems.

C-64 \$39.95

Xper

XPERS is the first "expert system" for the C-128 and C-64. While ordinary data base systems are good for reproducing facts, XPER can derive knowledge from a mountain of facts and help you make expert decisions. Large capacity. Complete with editing and reporting.

C-64 \$59.95

C-128 and C-64 are trademarks of Commodore Business Machines

Abacus Software



P.O. Box 7219 Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Call now for the name of your nearest dealer. Or to order directly by credit card, MC, AMEX of VISA call (616) 241-5510. Other software and books are available-Call and ask for your free catalog. Add \$4.00 for shipping per order. Foreign orders add \$12.00 per item. Dealer inquiries welcome-1200+ nationwide.

California) has converted its series of 3-D graphics adventure games for the ST: *King's Quest I*, *King's Quest II*, and *Walt Disney's The Black Cauldron*. The *King's Quest* games have been particularly popular on IBM computers.

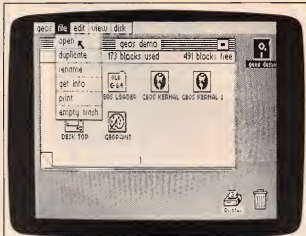
Spinnaker Software (Boston, Massachusetts) has converted *Fahrenheit 451*, *Treasure Island*, *Perry Mason*, *Nine Princes of Amber*, *Amazon*, and *Homework Helper Math* for the ST.

The popular eight-bit computers weren't ignored by software publishers, either, and some significant new programs were announced for the Commodore 64, 128, Apple, IBM PC/PCjr, and Atari. There isn't room here to list them all, but additional information can be found in this month's "News & Products" section.

One of the biggest hits at the show was a graphics-oriented operating system in final stages of development for the Commodore 64. Called *GEOS* (Graphic Environment Operating System), it's modeled after the user interface popularized by the Macintosh and adapted by the Atari ST and Amiga. *GEOS* loads from disk and replaces the 64's normal operating system. It speeds up disk access by a factor of five to seven times and displays a desktop screen with pull-down menus, icons, and windows. You manipulate these features by moving an on-screen pointer with a mouse or joystick.

Although some people were skeptical that *GEOS* could run fast enough on an eight-bit machine, even the unfinished version at CES ran surprisingly smoothly. By moving the pointer to a menu bar at the top of the screen and pressing the joystick button, you can select various options for managing files and running applications (see photo). To rename a file, for instance, you pull down the File menu and choose Rename. The new filename you enter appears on the directory or beneath the program's icon. To delete a file, you point to its icon, press the joystick button, and "drag" the icon to the trash can icon. To print a file, you drag its icon to the printer icon.

The screen would get cluttered



The desktop screen from *GEOS*, a remarkable Macintosh-like operating system for the Commodore 64. It offers pull-down menus, icons, windows, desk accessories, and custom type fonts, and also speeds up disk access five to seven times.

if the icons for every file were displayed at once, so *GEOS* lets you flip through windows as if they were pages in a book. The page number of the current window is displayed near its bottom margin. As the accompanying photo shows, *GEOS* uses the Commodore 64's high-resolution graphics screen to display a smaller-sized character set.

The desktop includes a full range of Macintosh-like desk accessories, such as a calculator, notepad, alarm clock, photo and text albums (for transferring data between applications), and a preferences window. The preferences window lets you adjust screen colors, change the shape and speed of the pointer, set the date and time, and so forth. The desk accessories can be called up while running other applications—if you're using the word processor and need to perform a calculation, for example, you can pop open the calculator, get your answer, close the calculator, and continue writing.

GEOS comes with two applications: *geoWrite*, a what-you-see-is-what-you-get word processor that lets you type on-screen with several different proportional fonts, and *geoPaint*, a graphics-drawing program with vertical and horizontal scrolling that lets you create images

as large as an 8½ × 11-inch page (80 dots per inch resolution).

The price for the entire package—*GEOS*, the desktop and desk accessories, *geoWrite* and *geoPaint*—is \$59.95. It was designed by Berkeley Softworks (Berkeley, California) and is scheduled to be available early this spring.

Timeworks (Deerfield, Illinois) is releasing three new programs for the Commodore 128 and one for the 64. *Partner 128* (\$59.95) and *Partner 64* (\$49.95) are desktop management programs similar to Borland International's *Sidekick* for the IBM PC. Both *Partner* programs include a multifunction business calculator, memo pad, appointment calendar, typewriter, address book, phone book, envelope addresser, and screen printer.

SwiftCalc 128 is a spreadsheet that takes advantage of the 128's 80-column mode. It has programmable keys, multilevel sorting, and several ways of charting data (including 3-D) that can be shown on the screen or printed out. Those who already own the original *SwiftCalc* for the Commodore 64 can upgrade to the 128 version for a nominal fee. Timeworks has also adapted *Sylvia Porter's Personal*

Financial Planner (\$69.95) for the 128's 80-column mode.

Cardco (Wichita, Kansas) announced a *Sidekick*-style product, too. Called *Side Saddle* (Cardco is negotiating with Borland for the *Sidekick* name), it offers quick access to a calculator, appointment calendar, telephone directory/dialer, memo pad, screen printer, and disk functions. It comes on a cartridge for the Commodore 64, with a 128 version to follow. Another interesting Cardco program is *Freeze Frame*, a screen printer that Cardco claims can capture any 64 screen on paper, even with commercial software. It works with any printer that emulates the Commodore 1525, and all Epson- and Okidata-compatibles. A 128 version of *Freeze Frame* is also planned.

The Commodore 128 got another boost when Cardco announced its Personal Productivity Series for the 128's CP/M mode. The first three products in the series are *Personal Accountant*, a financial manager for home or small business; *Personal Inventory*, for figuring your net worth; and *Personal Time Manager*, which can handle up to 26 events for as many as 240 people. Each program sells for \$39.95.

In addition to software, Cardco announced three hard disk drives for the 64 and 128. Available in late March for the 64 and shortly afterward for the 128, the drives will come in 5-, 10-, and 20-megabyte capacities for \$599.95, \$899.95, and \$1,299.95, respectively. They're capable of loading a full-function spreadsheet program in 2½ seconds.

Access Software (Salt Lake City, Utah) is releasing a number of products for the Commodore 64, 128, Atari, and Apple. The *Mach 128 Enhancement Package* is a cartridge and disk for the 64 and 128. When the cartridge is plugged in, it senses whether a 1541 or 1571 disk drive is connected and—in the case of the 128—which mode is active. Then it engages or disengages disk speed-up routines as appropriate. Typically there's a fivefold increase in loading speed with either the 1541 or 1571. The cartridge also has a system reset switch. In addition, the *Mach 128* disk includes a disk organizer utility, two machine language monitors, and a program that

expands BASIC workspace by 4K. The price is \$49.95.

The *Development System* (\$79.95) is a professional macro assembler and text editor for the Commodore 128 (128 or 64 mode) and 64. It includes *Spritemaster*, a utility for creating and animating sprites with machine language programs.

Leader Board, *The Pro Golf Simulator* is a 3-D game that offers a perspective view of the golf course. It's one of the most detailed sports simulations we've ever seen, with multiple 18-hole courses, handicapping, a wide range of clubs, and numerous other variables. The Commodore 64 version should be available immediately for \$39.95. *Inside Story—The Anatomy Learning System* is an educational program with 50 high-resolution graphics screens that let you explore the inner working of the human body. For the Commodore 64, the price is \$34.95.

For the Atari 400/800, XL, XE, and Apple II series, Access Software is releasing *Raid Over Moscow* and *Beach-Head II: The Dictator Strikes Back*, popular games previously available for the Commodore 64 and 128. In *Raid Over Moscow*, the Soviets have launched a nuclear attack on North America; your job is to deploy stealth bombers from an orbiting space station to destroy the warheads before they hit. It requires at least 48K RAM and sells for \$34.95. *Beach-Head II* is the sequel to *Beach-Head* and features speech synthesis, multiple screens, and the choice of playing another person or the computer. It requires at least 48K RAM and sells for \$39.95.

Access has signed an agreement with Multibotics, Inc. (Woods Cross, Utah) to market its line of robotic construction sets. The sets—for youngsters and adults—contain snap-together connectors, gears, shafts, clutches, wheels, electric motors, and other parts that make it possible to build all sorts of motorized contraptions that can be controlled by a personal computer. You can also experiment with digitized speech or temporarily turn a computer into a voltmeter or oscilloscope. Four different Multibot sets are available from \$39.95 to \$199.95. Interfaces are ready for the Commodore 64 and 128, and Access is working on interfaces for Atari, Apple, Amiga, and IBM computers.

If you've got a good memory, you might recall reading some reviews about a year ago of a Commodore 64 word processor called *SkiWriter*. Although the reviews were good, marketing problems kept the program from appearing on store shelves. Now it's been acquired by a British company, Mastertronic (U.S. offices in Frederick, Maryland). Two changes were made—the built-in telecommunications feature was dropped, and the program is being sold on disk instead of cartridge—but the price has been chopped from \$69.95 to \$15. There's also a Commodore 128 and Apple II version. At the same time, Mastertronic is introducing two more programs for the Commodore 64: *Basic3*, a spreadsheet, and *Instant Recall*, a filer that stores up to 30,000 characters of data. Both of these products sell for \$15, too.

Mindscape (Northbrook, Illinois) is bringing out *The Luschter Profile* (Apple, Commodore 64, IBM PC, and Mac), which constructs a psychological profile based on a person's reactions to colors and shapes; *The American Challenge: A Sailing Simulation* (Apple and IBM), an unusual America's Cup simulation that can be played by two people in remote locations using computers and modems; *Dick Francis' High Stakes* (Apple and IBM), an interactive text adventure that puts you in the role of a wealthy English horse owner; a talking Macintosh version of *Racter*, the AI (artificial insanity) program that holds bizarre conversations with humans; and *Stephen King's The Mist and James Bond: A View To A Kill* (Apple, IBM, and Mac), text adventures based on popular thrillers. All of these programs are \$39.95, except for *Racter*, which is \$44.95.

And finally, if you can spare \$39.95, you can now indulge any Rambo fantasies you might have with a Mindscape program called *Rambo: First Blood Part II*. But Rambo isn't the shoot-em-up action game you might expect—it's a text adventure. One of its features is a sophisticated parser that lets you communicate in plain English (which is more than the movie character Rambo can do). It runs on the Apple, IBM, and Macintosh. ☐

Tug-A-War

Mark Tuttle, Submissions Reviewer

Don't be fooled by the apparent simplicity of this two-player strategy game. It looks easy on the surface, but it's a stiff test of your concentration and ability to think ahead. The original version was written for the Commodore 64. We've added new versions for the Atari 400/800, XL, and XE, Apple II-series computers, Atari 520ST, Amiga, IBM PC/PCjr, and the TI-99/4A. Since the game is based on colors, every version requires a color monitor or TV. The IBM version requires BASICA and a color/graphics adapter for the PC or Cartridge BASIC for the PCjr. The Atari version requires at least 16K of RAM, and the Amiga version requires at least 512K.

Nearly everyone has played tug of war at one time or another. The traditional game pits two players or teams at opposite ends of a rope. At the middle of the rope is a flag, and each side tries to pull the flag into its territory. "Tug-a-War" is based on a similar concept. In this version, the flag is replaced with a round ball shape, and each player tries to maneuver the ball onto his or her side of the screen. Like many two-player games, the difficulty of Tug-a-War depends somewhat on the intelligence of your opponent. But even at the simplest level, you'll find that skill and foresight are essential to success.

Type in and save the appropriate program below. The rules are the same for every version (except Atari 520ST—see special instructions).

Battle Of The Colors

When you run Tug-a-War, two sets of colored boxes appear, one above the other. The lower, longer series of squares is the playing field. Near the middle of the playing field area is a

round ball; the outermost boxes at each end of the playfield represent each player's home position. The players alternate turns, each trying to move the ball in their own direction, until it reaches one of the home squares.

So far, so good—but how do you move the ball? It's done not by pulling a rope, but by changing the colors of boxes in the playfield. The color of the square under the ball determines which direction it moves and how far it travels. On any given turn, the ball can move either one or two squares to the left, or one or two squares to the right. At the top of the screen are four boxes that show you which colors are linked to which directions. For instance, the leftmost box shows you which color makes the ball move one square to the left. The next box to the right shows you which color makes it move two squares to the left. The second pair of boxes show you which colors make the ball move in the opposite direction, to the right. By changing the color of the box where the ball is currently located, you can make it move toward your home square.

The playfield contains 11 boxes (9 in the TI version, 10 in the Atari ST version). When the game begins, each of these boxes is randomly given one of the four colors shown at the top of the screen. On each turn, you may change the color of one, several, or all of the boxes (however, you must always change at least one box). Below each box is a number which represents its distance from the home position of the player whose turn it is. For instance, if you are the player on the left, then on your turn the boxes are numbered 1, 2, 3, etc., from left to right (the tenth box is marked with a 0, and the eleventh

with an A). When it's the right player's turn, the numbering is reversed (the rightmost box is 1, etc.).

To take a turn, you must select a number that corresponds to the numbers shown below the boxes in the playfield. This is done by pressing a single key. Press a number key from 1-0 to select one of the first ten values, or press the A key to choose the eleventh box. The number you choose determines how many boxes change color. For instance, if you press 1, only one box (the one nearest your home square) changes color. If you press 2, the two boxes nearest your home box change, and so on.

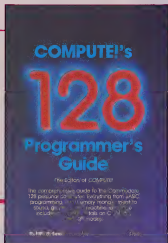
Where do the new colors come from? Every box cycles through the same series of four colors shown in the uppermost set of boxes, going from left to right. For example, if the colors shown there are white-blue-red-purple (the exact colors may be different on your computer), then a white square always changes to blue; a blue square always changes to red; a purple square changes to white, and so on. In other words, the box's current color determines which color it gets after the next color change.

Though every turn involves at least one color change, the ball doesn't necessarily move on every turn. It only moves when you change all the boxes between your home position and the current position of the ball. For example, if the ball is three boxes away from your home square, then you must change the color of at least three boxes in order to move it at all.

Foresight Rewarded

As you can see, there's much more to this game than appears on the surface. At first you might be tempted to try to move the ball as

COMPUTE!'s PROGRAMMER'S GUIDES



COMPUTE!'s 128 Programmer's Guide

ISBN 0-87455-031-9

Editors of COMPUTE! 464 pages

Written and compiled by the most technically proficient authors in consumer computing today, the technical staff of COMPUTE! Publications, this guide to the powerful Commodore 128 computer contains a wealth of information for every programmer. Explore BASIC 7.0 through countless hands-on examples and sample programs. Learn how to create dazzling graphics and sophisticated sounds in both BASIC and machine language. See how to program peripherals, such as disk drives, printers, and modems. Enter the world of CP/M, just one of the three modes of the 128. There are even chapters introducing you to machine language programming and the computer's method of managing memory. *COMPUTE!'s 128 Programmer's Guide* includes numerous appendices covering everything from error messages to memory maps.

\$16.95

**Look for these new books at a
bookstore or a computer store near you.
Or order directly from COMPUTE! Books.
Call toll-free 1-800-346-6767. In NY call
212-887-8525.**

***Everything you need for successful,
entertaining, and challenging
programming on your Amiga, Atari
ST, or Commodore 128 computer.***

***Each book is carefully written in
COMPUTE!'s lively, understandable
style to help even beginning
programmers learn the ins and outs
of their personal computers.***

COMPUTE!'s ST Programmer's Guide

0-87455-023-8

Editors of COMPUTE!

Complete and comprehensive, yet easy to understand, *COMPUTE!'s ST Programmer's Guide* is a must-buy for any Atari ST owner. The technical staff of COMPUTE! Publications has put together a reference guide to programming that takes the reader through every aspect of this newest Atari personal computer. Logo and BASIC, the two programming languages now available for the machine, are explored in detail. From programming concepts to writing programs, the scores of ready-to-type-in examples show just what can be done, and how to do it. Advanced features of this new-generation computer, such as GEM and TOS, the ST's user interface and operating system, are illustrated as readers write their own applications. Valuable appendices provide information programmers need, including GEM VDI opcodes and a list of ST resources.

\$16.95

COMPUTE!'s Amiga Programmer's Guide

0-87455-028-9

Editor

Covering AmigaDOS, BASIC, intuition, C, machine language, and the other important programming tools which accompany the new Amiga, *COMPUTE!'s Amiga Programmer's Guide* is a clear and thorough guide to the inner workings of this fascinating, new-generation computer. The great speed of its 68000 microprocessor, coupled with the versatility of the Amiga-specific graphics and sound chips, makes the Amiga one of the most powerful computers available today.

\$16.95

COMPUTE! Publications, Inc. 
One of the ABC Publishing Companies

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

often as possible. But that's usually a losing strategy. Remember, the direction the ball moves depends on the color of its square before you take the turn.

In many cases, you'll want to move the ball only if it's on a color that moves it toward your goal. But like other games of strategy, Tug-a-War rewards the player who looks beyond the current move and tries to set things up for future moves; sometimes it's wise to make a small, temporary sacrifice in order to benefit later in the game. Because the boxes change colors in the same sequence, the effect of your own move is always completely predictable. However, since a single turn can change the color of many boxes, dramatic changes of fortune are also possible.

Amiga And 520ST Versions

Since the mouse is an integral system feature on both the Amiga and ST, both of these versions substitute mouse input for keyboard input. To select a square, simply move the mouse pointer to the desired box and press the left mouse button. Because keyboard prompts are unnecessary, no numbers are displayed below the playfield boxes.

Before entering BASIC to load the ST version, you should switch to the low-resolution graphics mode (use the Set Preferences option in the desktop's Options menu). Also, if your ST has 512K and a disk-based operating system, before running the program you should turn off buffered graphics (controlled by the Buffer Grph option in the Settings menu; it's off when no check appears beside the option in the menu). The standard 520ST leaves only about 5K free for BASIC programs, so Tug-a-War won't fit into memory unless the buffered graphics option is turned off. The program fits with buffered graphics switched on only if you have a 1040ST, or a 520ST with memory expansion, or a 520ST that has been upgraded with the TOS operating system in ROM chips (Read Only Memory).

The Amiga version uses the computer's built-in speech feature to announce the players' turns. In other respects, these games work exactly like the others.

For instructions on entering these listings, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.



"Tug-a-War" for the Commodore 64 and 128 is a game that looks simple, but demands great concentration and foresight.

Program 1: Tug-A-War For Commodore 64/128

```

JJ 100 POKES32B0,0:PRINT"[CLR]
      :BC=532B1:POKEBC,5:PC
      (1)=5:PC(2)=7:PS=6:CB=1
      :PL=1:X=20
CE 110 B=55715:AS="OEV3P[DOWN]
      [3 LEFT][E]3[DOWN]
      [3 LEFT][L8P30]:PS(1)="G
      REEN[RV5][BLK][OFF]
      [LEFT]:PS(2)="YELLOW":
      BS="2 UP"
DM 120 TMS="HOME][9 DOWN]:QS
      ="[RV5][BLK][34 SPACES]
      ":DIMCL(11)
QQ 130 FOR=0T023:PRINT"[RV5]
      [BLK][39 SPACES]"
KE 140 POKEL063+(40*1),160:POK
      E55335+(40*1),0:NEXT
AQ 150 PRINT"[RV5][39 SPACES]
      [HOME]:POKE2023,160:PO
      KE56295,0
DO 160 PRINT"[HOME]"SPC(15)"
      [RV5]TUG-A-WAR":PRINTSP
      C(13)*[DOWN][RV5][WET]
      ASB$(RED)"ASB$(BLK)
      [CYN]"ASB$(PUR)"ASB$
RS 170 PRINTSPC(14)*[BLK]1"SPC
      (2)*2"SPC(3)*1"SPC(2)*2
      [DOWN]:PRINTSPC(15)"
      [RV5]<C"SPC(5)"C":
HK 180 GOSUB620:PRINTTMS"
      [DOWN][RV5][GRN]"ASB$
      :FOR2=1T011:Y=INT(4*RD
      (1))+1:CL(2)=Y:POKE646,
      Y
FQ 190 PRINT"[RV5]"ASB$:NEXT
CH 210 POKEL646,PEEK(B)AND15:PR
      INTMS"[2 DOWN]"SPC(X)"
      [RV5]Q":POKEBC,PC(PL)
SP 220 PRINTMS"[14 DOWN]"SPC(
      16)*[RV5][BLK]
      [3 SPACES]"
HR 230 AN=0:PRINTMS"SPC(7)"
      [18 DOWN][BLK][RV5]HOW
      [SPACE]MANY TO CHANGE
      [OFF]"PS(PL)
MA 240 PRINTSPC(17)*[DOWN]
      [RV5](1-A)[2 DOWN]
      [3 LEFT][E]3[LEFT]:
BR 250 POKEL204,0:POKE19B,0:WAI
      T190,1:GETMTS
RB 260 IFASC(MTS)<40BASC(MTS)>57T
      HEN2
50
XR 270 POKEL204,1:IFMTS="A"THEN
      AN=11:MTS="LEFT[AL]:G

```

```

OTO300
CR 200 IFMTS="8"THENAN=10:MTS=
      "10":GOTO300
HC 290 AN=VAL(MTS)
KM 300 PRINT"[2 LEFT]
      [5 SPACES][3 LEFT]"MTS
7FS 310 IFAN<10RAN>11THEN220
PF 320 IFPL=2THENAN=12-AN:GOTO
      440
HE 330 IFAN<PSTHENCK=1
KQ 340 FORQ=1TOAN:IFCL(Q)=4THE
      NCL(Q)=1:GOTO360
GE 350 CL(Q)=CL(Q)+1
KJ 360 NEXTQ:PRINTTMS"PRINT"
      [BLK][RV5][GRN]"ASB$:
      FOR2=1T011:POKE264,CL(2
      )":PRINT"[RV5]"ASB$:
QD 370 NEXT2:PRINT"[YEL][RV5]"
      AS:POKE646,PEEK(B)AND15
      :PRINT"[HOME][11 DOWN]"
      SPC(X)"[RV5]"
FH 380 IFCK=1THENCK=0:GOTO400
MH 390 CNPEK(B)AND15GOSUB490,
      500,510,520
BF 400 IFPS<1THENPL=1:WC=5:B=5
      569B:X=2:GOTO530
DP 410 IFPS>11THENPL=2:WC=7:B=5
      5734:X=30:GOTO530
KD 420 IFPL=1THENPL=2:GOSUB640
      :GOSUB630:GOTO510
HE 430 PL=1:GOSUB640:PRINT"
      [10 UP]":GOSUB620:GOTO2
      10
XH 440 FORQ=ANT011
DH 450 IFAN<10RAN>11THEN220
DE 460 IFAN<PSTHENCK=1
GJ 470 IFCL(Q)=4THENCL(Q)=1:GO
      T0360
HB 480 CL(Q)=CL(Q)+1:GOTO360
JX 490 B=B+6:X=X+6:PS=PS+2:RET
      URN
PA 500 B=B+3:X=X+3:PS=PS+1:RET
      URN
XM 510 B=B+6:X=X+6:PS=PS+2:RET
      URN
DE 520 B=B+3:X=X+3:PS=PS+1:RET
      URN
PF 530 POKEL646,PEEK(B)AND15:PR
      INT"[HOME][11 DOWN]"SPC
      (X)"[RV5]Q":PRINTTMS"
      [10 DOWN]"
PG 540 FOR2=1T05:PRINTQ3:NEXT:
      GOSUB640:GOSUB640
FM 550 PRINT"[HOME][11 DOWN]"S
      PC(11)P$(PL)"IS THE WI
      NNER":2=WC:FOR1=1T011:P
      OKEND,2
CP 560 IF2=0THEN2=WC:GOSUB610:
      NEXT
SJ 570 2=0:GOSUB610:NEXT
BP 580 POKESC,15:PRINTTMS"SPC(1
      1)"[10 DOWN][RV5]LIKE T
      O PLAY AGAIN? 2 DOWN]
      [11 LEFT][RV5]Y/A"
RA 590 POKEL19B,0:WAIT19B,1:GET
      MTS:IFMTS<0:"B"THENRAN
      JC 600 POKEL190,0:RYS190
QC 610 FORP=1T0200:NEXTP:RETUR
      N
CG 620 PRINTMS"[5 DOWN][RV5]
      [BLK][2 SPACES]<C 1
      [2 SPACES]2[2 SPACES]3
      [2 SPACES]4[2 SPACES]5
      [2 SPACES]6[2 SPACES]7
      [2 SPACES]8[2 SPACES]9
      [2 SPACES]0[2 SPACES]A"
      :RETURN
AA 630 PRINTMS"[5 DOWN][RV5]
      [BLK][2 SPACES]A
      [2 SPACES]0[2 SPACES]9
      [2 SPACES]8[2 SPACES]7
      [2 SPACES]6[2 SPACES]5

```

From the publishers of *COMPUTE!*



April 1986 *COMPUTE!* Disk

All the exciting programs from the past three issues of *COMPUTE!* are on one timesaving, error-free floppy disk that is ready to load on your Commodore 64 and 128 computers. The April 1986 *COMPUTE!* Disk contains the entertaining and useful Commodore 64 and 128 programs from the February, March, and April 1986 issues of *COMPUTE!*.

The April 1986 *COMPUTE!* Disk costs \$12.95 plus \$2.00 shipping and handling and is available only from *COMPUTE!* Publications.

For added savings and convenience, you may also subscribe to the *COMPUTE! Disk*. At a cost of only \$39.95 a year (a \$12.00 savings), you'll receive four disks, one every three months. Each disk will contain all the programs for your Commodore machine from the previous three issues of *COMPUTE!*

This is an excellent way to build your software library while you enjoy the quality programs from *COMPUTE!*.

Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

For more information or to order the February 1986 *COMPUTE! Disk*, call toll free 1-800-346-6767 (in NY 212-265-8360) or write *COMPUTE! Disk*, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE! Publications, Inc. 
One of the ABC Publishing Companies

```

[2 SPACES]4[2 SPACES]3
[2 SPACES]2[2 SPACES]1
[PRINT]C;"RETURN
AQ 640 SPACES" [5 DOWN] [RVS]
[BLK] [39 SPACES]:"RETUR
N

```



This version of "Tug-a-War" runs on all Atari 400, 800, XL, and XE computers.

Program 2: Tug-A-War For Atari 400/800, XL, XE

Version by Kevin Mykityn, Editorial Programmer

```

N 10 CB=PEEK(106)-B:POKE 10
4,CB-4:GRAPHICS 18:OIM
C(4),COL(11),MOV(4),
K(1):C="Z222"
N 20 MOV(1)=-1:MOV(2)=-2:MO
V(3)=1:MOV(4)=2:OPEN #
1,4,B,"K":GOSUB 190:G
OSUB 250
N 30 BP=#:PL=#
N 40 PL=(PL-1):GOSUB 290:PO
KE 53249,72+B:BP
N 50 GET #1,K:CHR$(K):IF
K="A" THEN SP=1:GOTO
0 90
N 60 IF K="B" THEN SP=10:G
OTO 90
N 70 IF K<"1" OR K>"9" TH
EN 50
N 80 BP=VAL(K)
N 90 IF (PL=0 AND 12-SP>BP)
OR (PL=1 AND SP<BP) T
HEN 120
N 100 BP=BP+MOV(COL(BP)):IF
BP>12 THEN BP=12
N 110 IF BP=-1 THEN BP=0
N 120 GOSUB 320
N 130 IF BP>0 AND BP<12 THE
N 40
N 140 POKE 53249,72+B:BP:PO
SITION 4,9
N 150 IF BP=0 THEN PRINT #6
:" BLUE WINS! ":GOTO
170
N 160 PRINT #6:" RED WINS!
"
N 170 POSITION 0,11:PRINT #
6:"(3 SPACES)PRESS AN
Y KEY(3 SPACES)"
N 180 GET #1,K:GOTO 30
N 190 NS=CB+256:POSITION 5,
5:PRINT #6:"PLEASE WA
IT"
N 200 FOR A=0 TO 1023:POKE
NS+A,PEEK(57344+A):NE
XT A
N 210 POKE 756,CB:FOR A=NS+
464 TO NS+471:POKE A,
126:NEXT A
N 220 FOR A=NS+1536 TO NS+1
920:POKE A,0:NEXT A:FO
R A=NS+1536 TO NS+16
83:POKE A+256,126:POK
E A+128,126:NEXT A

```

```

N 230 POKE 559,46:POKE 523,
1:POKE 53250,72:POKE
53249,168:POKE 53277,
3:POKE 54279,CB+4:POK
E 704,15:POKE 705,52
N 240 FOR A=NS+1536 TO NS+1
601:READ B:POKE A,B:N
EXT A:POKE 706,132:RE
TURN:DATA 24,0,0,0,2,
4
N 250 PRINT #6:"(CLEAR)":PO
SITION 4,0:PRINT #6:"
1 2 1 2":POSITION 6,1
:PRINT #6:"Z 2 2 2":P
OSITION 7,2:PRINT #6:
"CB CB"
N 260 GB=PEEK(560)+256:PEEK
(561):POKE GB+10,6:PO
KE GB+16,6
N 270 POSITION 4,6:FOR A=1
TO 11:Q=INT(RND(1)*4+
1):COL(A)=Q:PRINT #6:
C(Q,Q):NEXT A
N 280 POSITION 0,11:PRINT #
6:" (3 SPACES)PRESS AN
Y KEY(3 SPACES)":RETUR
N
N 290 POSITION 4,9
N 300 IF PL=1 THEN PRINT #6
:"BLUE'S TURN":POSITI
ON 4,5:FOR A=1 TO 9:P
RINT #6:A:NEXT A:PRI
NT #6:"0A":RETURN
N 310 PRINT #6:"RED'S TURN
":POSITION 4,5:PRINT
#6:"A0":FOR A=9 TO 1
STEP -1:PRINT #6:A:
NEXT A:RETURN
N 320 IF PL=0 THEN 340
N 330 FOR A=1 TO SP:GOSUB
350:NEXT A:RETURN
N 340 FOR A=11 TO 12-SP:ST
EP -1:GOSUB 350:NEXT A
:RETURN
N 350 COL(A)=COL(A)+1-4*(CO
L(A)=4):POSITION 3+A,
6:PRINT #6:C(COL(A),
COL(A)):RETURN

```



"Tug-A-War" for the IBM PC and PCjr.

Program 3: Tug-A-War For IBM PC/PCjr

Version by Kevin Mykityn, Editorial Programmer

```

N 10 GOTO 30
N 20 FOR ROW=0 TO 2:LOCATE Y+RO
W,PS+3:X:PRINT B:NEXT:RET
URN
N 30 KEY OFF:SCREEN 0,0:WIDTH 4
0:B=CHR$(222)+CHR$(219)+C
HR$(221)+CHR$(31)
N 40 DIM COL(11):BP=6:C(1)=2:C
(2)=6:C(3)=3:C(4)=5:PL=0:RA
NDIZE TIMER
N 50 MOV(1)=-1:MOV(2)=-2:MOV(3)
=1:MOV(4)=2
N 60 GOSUB 190

```

```

N 70 PL=(PL=0):GOSUB 230:GOSUB
240:GOSUB 180
N 80 KS=INKEY$:IF K="a" OR K="
" THEN SP=11:GOTO 110
N 90 IF K="B" THEN SP=10:GOTO
110
N 100 IF K<"1" OR K>"9" THEN
BP ELSE SP=VAL(K)
N 110 IF (PL=0 AND 12-SP>BP) OR
(PL=1 AND SP<BP) THEN 1
30
N 120 BP=BP+MOV(COL(BP)):IF BP>
12 THEN BP=12 ELSE IF BP=
-1 THEN BP=0
N 130 GOSUB 260:IF BP>0 AND BP<
12 THEN 70
N 140 GOSUB 230:LOCATE 20,15:CO
LOR 9:IF BP=0 THEN PRINT
" Blue Wins! ":GOTO 160
N 150 COL 4:PRINT " Red Wins!
"
N 160 LOCATE 23,8:COLOR 14:PRIN
T " Press any key to play
again":GOSUB 180
N 170 AS=INKEY$:IF AS="" THEN 1
70 ELSE RUN
N 180 DEF SP=0:POKE 1050,PEEK
(1027):RETURN
N 190 C,Q,Y=3:X=0:FOR A=1 TO 4:
PS=4+A:COL 4:PRINT #6:GOSUB
20:NEXT
N 200 COL 14:PRINT SP(15):CHR
$(17)" "STRING$(2,17)" "
CHR$(16)" "STRING$(2,16)
N 210 Y=13:X=1:COL 9:PS=0:GOS
UB 20:FOR A=1 TO 11:Q=
INT(RND(1)*4+1):C(1)=Q:COL
OR C(Q):PS=A:GOSUB 20:NEXT
T:COL 4:PS=12:GOSUB 20
N 220 LOCATE 23,8:COLOR 10:PR
INT " Press (0-9) or 'A' f
or all":RETURN
N 230 COL 15:LOCATE 14,8:PS+2
:PRINT CHR$(219):RETURN
N 240 LOCATE 20,15:IF PL=0 THEN
COL 4:PRINT "Red's Tur
n":LOCATE 11,5:PRINT "A
0":FOR A=9 TO 1 STEP-1
:PRINT A:NEXT:RETURN
N 250 COL 9:PRINT "Blue's Tur
n":LOCATE 11,4:FOR A=1 TO
9:PRINT A:NEXT:PRINT "
0 A":RETURN
N 260 IF PL=0 THEN 280
N 270 FOR A=1 TO SP:GOSUB 290:N
EXT:RETURN
N 280 FOR A=11 TO 12-SP STEP-1:
GOSUB 290:NEXT:RETURN
N 290 COL(A)=COL(A)+1-4*(COL(A)
=4):COL 4:COL(A):PS=A:
GOSUB 20:RETURN

```

Program 4: Tug-A-War For Apple

Version by Tim Victor, Editorial Programmer

```

N 100 GOSUB 400
N 110 H=0:R=0
N 120 HCOL=0:3:FOR TD=-1 TO
0 STEP 2:FOR TN=0 TO
1:FOR TX=-1 TO TN STEP
2:GOSUB 500:NEXT:1:NEXT
NEXT
N 130 VP=40
N 140 FOR I=0 TO 3:HC=CT(1):
PS=4.5+1:GOSUB 430:
NEXT
N 150 VP=146:HC=1:PS=0:G
OSUB 430:GOSUB 460
N 160 FOR I=0 TO 10:BC(1)=1
NT(&RND(1)*84):HC=CT
(BC(1)):PS=I+1:GOSUB
430:NEXT

```


From the publishers of *COMPUTE!*



April 1986 *COMPUTE!* Disk

All the exciting programs from the past three issues of *COMPUTE!* are on one timesaving, error-free floppy disk that is ready to load on your IBM PC and PCjr computers. The April 1986 *COMPUTE!* Disk contains the entertaining and useful IBM programs from the February, March, and April 1986 issues of *COMPUTE!*

The April 1986 *COMPUTE!* Disk costs \$12.95 plus \$2.00 shipping and handling and is available only from *COMPUTE!* Publications.

For added savings and convenience, you may also subscribe to the *COMPUTE!* Disk. At a cost of only \$39.95 a year (a \$12.00 savings), you'll receive four disks, one every three months. Each disk will contain all the programs for your IBM machine from the previous three issues of *COMPUTE!*

This is an excellent way to build your software library while you enjoy the quality programs from *COMPUTE!*

Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

For more information or to order the February 1986 *COMPUTE!* Disk, call toll free 1-800-346-6767 (in NY 212-265-8360) or write *COMPUTE!* Disk, P.O. Box 5038, F.D.R. Station, New York, NY 10150.



Apple "Tug-A-War," a challenging strategy game.

```

C1 170 HC = 6:PS = 12: GOSUB 430
C2 180 BP = 5: GOSUB 470
C3 190 VTAB 21: FOR I = 1 TO 11:
HTAB I * 3 + 2: IF I < 1
0 THEN PRINT CHR$ (48 + I)
:
C4 200 IF I = 10 THEN PRINT "0";
C5 210 IF I = 11 THEN PRINT "A";
C6 220 NEXT : VTAB 23: HTAB 1: P
RINT "GREEN'S MOVE!";
C7 230 GOSUB 520:A = A - 1: IF (
BP < A) THEN BP = BP +
JT(BC(BP))
C8 240 FOR I = 0 TO A:BC(1) = BC
(1) + 1 - 4 * BC(1) = 3)
:HC = CT(BC(1)):PS = I +
1: GOSUB 430: NEXT
C9 250 GOSUB 470
C10 260 IF BP < 0 OR BP > 10 THEN
360
C11 270 VTAB 21: FOR I = 1 TO 11:
HTAB (12 - I) * 3 + 2: I
F I < 10 THEN PRINT CHR$
(48 + I);
C12 280 IF I = 10 THEN PRINT "0";
C13 290 IF I = 11 THEN PRINT "A";
C14 300 NEXT : VTAB 23: HTAB 1: P
RINT "BLUE'S MOVE: ";
C15 310 GOSUB 520:A = 11 - A: IF
(BP > A) THEN BP = BP +
JT(BC(BP))
C16 320 FOR I = 10 TO A STEP - 1:
BC(1) = BC(1) + 1 - 4 * (
BC(1) = 3):HC = CT(BC(1)):
:PS = I + 1: GOSUB 430: N
EXT
C17 330 GOSUB 470
C18 340 IF BP < 0 OR BP > 10 THEN
360
C19 350 GOTO 190
C20 360 PS = 12 * (BP > 0) - 1: H
COLOR= 4 * (BP > 0): GOSU
B 490
C21 370 VTAB 23: HTAB 1: IF BP <
0 THEN PRINT "GREEN WINS
": GOTO 390
C22 380 IF BP > 10 THEN PRINT "BL
UE WINS "
C23 390 GET AS: GOTO 110
C24 400 FOR I = 0 TO 3: READ CT(I
): NEXT
C25 410 FOR I = 0 TO 3: READ JT(I
): NEXT : RETURN
C26 420 DATA 3,5,6,2,-1,-2,1,2
C27 430 HCOLOR= HC: FOR VP = VP T
O VP + 10
C28 440 HPLLOT PS * 21 + 1,VP TO P
S * 21 + 17,VP: NEXT
C29 450 RETURN
C30 460 HCOLOR= 3: FOR YP = VP +
1 TO VP + 9 STEP 2: HPLLOT
PS * 21 + 1,YP TO PS * 2
1 + 17,YP: NEXT : RETURN
C31 470 IF BP < 0 OR BP > 10 THEN
RETURN

```

```

I5 480 HCOLOR= 4 * (CT(BC(BP)) >
3):PS = BP
I6 490 FOR YP = VP + 3 TO VP + 7
: HPLLOT PS * 21 + 27,YP T
O PS * 21 + 32,YP: NEXT :
RETURN
I7 500 TP = 124 + (TD + TN) * 21
+ TN * TX * 4:TL = TP +
TD * 3:TR = TP - TD * 3
I8 510 HPLLOT TR,60 TO TL,57 TO T
R,54: RETURN
I9 520 POKE 49160,0: GET AS: IF
AS = CHR$(3) THEN END
I10 530 IF AS = CHR$(3) THEN END
I11 540 IF AS < "A" AND AS < ">
" * AND AS < "0" OR AS >
"9") THEN 520
I12 550 IF AS = "A" OR AS = "a" T
HEN AS = CHR$(59)
I13 560 IF AS = "0" THEN AS = CHR
$(58)
I14 570 A = ASC (AS) - 48: RETURN

```

Program 5: Tug-A-War For Atari ST

Version by Kevin Mykityn, Editorial Programmer

```

10 fullw 2:clearw 2:color 1,1,1
20 bp = 6:cl = 6:cl2 = 7:cl3 = 10:cl4 = 12:pl
= 0
30 mov(1) = -1:mov(2) = -2:mov(3) = 1:mov(
4) = 2
40 gosub 170:gosub 270
50 pl = (pl - 0):gosub drawball:gosub play
er
60 gosub readmouse:if y < 96 or y > 127 o
r x < 34 or x > 273 then 60
70 sp = int(x - 11)/24
80 if (pl = 0 and sp > bp) or (pl = -1 and sp
< bp) then 110
90 t = c(col(bp):color 1,1,1:pcircle bp*24 +
2,2,91,6
100 bp = bp + mov(col(bp)):if bp > 11 the
n bp = 11 else if bp = -1 then bp = 0
110 gosub colchange
120 gosub drawball:if bp > 0 and bp < 11 th
en 50
130 gotoxy 13,14:if bp = 0 then color 5:pr
nt "Blue Wins!":goto 150
140 color 2:print "Red Wins!"
150 gotoxy 10,16:color 1:print "Press Mou
se Button":gosub readmouse:clear go
to 10
160 drawball:color 1,1,1:pcircle bp*24 + 2
,2,91,6:return
170 for a = 75 to 105 step 30:linef 10,a,298,
a:next
180 for a = 10 to 298 step 24:linef a,75,a,10
5:next
190 color 1,5:fill 12,77:color 1,2:fill 296,77
200 gotoxy 12,3:print "1 2 1 2"
210 for a = 100 to 220 step 24:linef a,13,a,3
7:next
220 for a = 19 to 37 step 18:linef 100,a,148,
a:linef 172,a,220,a:next
230 color 1,6:fill 101,20:color 1,7:fill 125,20
240 color 1,10:fill 173,20:color 1,12:fill 197,
20
250 gotoxy 13,4:print chr$(40)," ",chr$(3)
260 return
270 for a = 1 to 10:q = int(rnd(1)*4 + 1):col(a
) = q:color 1,c(q):fill 25 + a*24,77
280 next:return
290 readmouse:poke contrl,124
300 poke contrl + 2,0:poke contrl + 6,0
310 v:delay(0:if peek(intout) = 0 then 310
320 x = peek(ptsout):y = peek(ptsout + 2)
330 return

```

```

340 colchange: if pl = 0 then 360
350 for a = 1 to sp:gosub 370:next:return
360 for a = 10 to sp step 1:gosub 370:next:
return
370 col(a) = col(a) + 1 + 4*(col(a) = 4)
380 color 1,c(col(a)):fill 25 + a*24,77
390 return
400 player: gotoxy 13,14:if pl = 0 then colo
r 2:print "Red's Turn ":return
410 color 5:print "Blue's Turn":return

```



Use the mouse to play the Atari ST version of "Tug-A-War."

Program 6: Tug-A-War For TI-99/4A

Version by Patrick Parrish, Programming Supervisor

```

100 GOTO 150
110 FOR I = 1 TO LEN(AS)
120 CALL HCHAR(R,C+1,ASC(SE
G$(AS,I,1)))
130 NEXT I
140 RETURN
150 RANDOMIZE
160 CALL COLOR(14,1,7)
170 CALL SCREEN(2)
180 PC(0) = 5
190 PC(1) = 7
200 PS(0) = "BLUE"
210 PS(1) = "RED"
220 Y$(0) = "< 1 2 3 4 5
6 7 8 9 "
230 Y$(1) = " 9 8 7 6 5
4 3 2 1 -> "
240 KHAR(0) = 0
250 KHAR(10) = 5
260 FOR I = 96 TO 136 STEP 8
270 CALL CHAR(I,"0000000000
000000")
280 CALL CHAR(I+1,"0F0F0F0F
0F0F0F0F")
290 CALL CHAR(I+2,"3078FCFC
FC7830")
300 CALL CHAR(I+3,"00103010
101030")
310 CALL CHAR(I+4,"003B4404
0010207C")
320 NEXT I
330 PS = 5
340 PL = 0
350 BP = 17
360 CALL CLEAR
370 GOSUB 1000
380 PRINT TAB(11),"TUG-A-WA
R"
390 PRINT :
400 BS = CHR$(128):CHR$(128) &
CHR$(129)
410 PRINT TAB(9),"hhippq xx
y":BS
420 PRINT TAB(9),"hk1ptq x(
y)":CHR$(128):CHR$(132):
CHR$(129)
430 PRINT TAB(9),"hhippq xx
y":BS

```

COMPUTE!'s All New Apple Applications Special

COMPUTE!'s latest *Apple Applications Special* features in-depth articles and interviews, all the inside news about Apple, clearly written tutorials, software buyer's guides, new product information, and valuable ready-to-type-in programs for all Apple users.



Apple owners find these special Apple issues the most understandable, complete, and valuable resources available today.

PLUS

All the programs in *COMPUTE!'s Apple Applications Special* are also available on a timesaving disk, ready to run on your Apple II, II+, IIe, and IIfx. The Disk costs only \$12.95 (plus \$2.00 shipping and handling) and gives you immediate access to all the great programs in this special issue.

Look for the Spring/Summer 1986 issue of *COMPUTE!'s Apple Applications Special* on sale where you buy other *COMPUTE!* publications, or order directly from *COMPUTE!*. This special issue goes on sale April 8, 1986.

Send in the attached order card or call toll free 800-346-6767 (in NY call 212-887-8525).

Features

• Business Applications

'86 Apple: An Interview with John Sculley

A wide-ranging interview with the president of Apple. The company's plans for the coming year, its markets, the new Macintosh, and the viability of the Apple II.

Business Software Buyer's Guide

A buyer's guide to the newest Apple II and Macintosh word processors, databases, spreadsheets, and more.

The Expanding Mac

• Education

Apple Rules the Schools

Why does Apple have a lock on educational computing? Comments from teachers, administrators, and Apple.

Computers and the Humanities

Educational Software Buyer's Guide

• The Expanding Apple

It's New II

A multitude of new hardware and software for the Apple II line—from color printers to Mac-like software—is evaluated.

Weirdware: Off the Beaten Software Path

Weirdware—out of the ordinary software—can turn the Apple II or Macintosh into a telescope, astrological fortune-teller, baby evaluator, and much, much more.

MacAdd: More for the Macintosh

Applications

• Utilities and Tutorials

Windows

Create Macintosh-style windows on any Apple II-series computer. Set window size, open, close, and retrieve information.

Mouse Cursor

A Macintosh BASIC utility for altering the mouse pointer. Design data can be saved, then used in other BASIC programs.

Your Personal Ledger

A complete personal financial application for tracking expenses, income, and assets. Easy to use, and packed with features from report generation to customized category codes.

Personal Publishing With Your Macintosh

Tutorial and guide to using such software as *MacPaint* and *MacWrite* to customize letterheads, cards, banners, and more.

Keynote

• Education and Recreation

Lexitron

Entertaining word game where players try to beat the clock while finding as many hidden words as possible.

Backgammon

Play the computer in this classic game. This version observes all the rules of standard backgammon.

Apple Automatic Proofreader

COMPUTE! Publications, Inc. 

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
625 7th Avenue, 6th Floor, New York, NY 10019

Business of COMPUTE!: COMPUTE!'s Gazette • COMPUTE!'s Gazette Disk • COMPUTE!'s Books • COMPUTE!'s Apple Applications

```

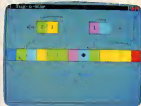
440 PRINT
450 PRINT TAB(11); "<-"; TAB(
18); "->"
460 FOR I=1 TO 15
470 PRINT
480 NEXT I
490 FOR I=1 TO 9
500 RANDOMIZE
510 KHAR(I)=INT(4*RND)+1
520 NEXT I
530 FOR R=1 TO 15
540 CALL HCHAR(R,2,96,2)
550 FOR I=1 TO 9
560 KH=96+KHAR(I)*8
570 CALL HCHAR(R,183+1,KH)
580 CALL HCHAR(R,183+2,KH)
590 CALL HCHAR(R,183+3,KH+1)
600 NEXT I
610 CALL HCHAR(R,31,136,2)
620 NEXT R
630 CALL HCHAR(14,BP,96+KHA
R(P),88+2)
640 IF (PS=0)+(PS=10) THEN 1
100
650 A$=V$(PL)
660 R=17
670 C=1
680 GOSUB 110
690 CALL HCHAR(24,17,32)
700 A$=P$(PL)&"'S TURN "
710 R=20
720 C=11
730 GOSUB 110
740 R=22
750 C=14
760 A$="(1-9)"
770 GOSUB 110
780 GOSUB 1020
790 CALL KEY(0,K,H)
800 IF H=0 THEN 790
810 IF (K<49)+(K>57) THEN 79
0
820 AN=K-48
830 CALL HCHAR(24,17,K)
840 IF PL=0 THEN 890
850 AN=10-AN
860 S=AN
870 E=9
880 GOTO 910
890 S=1
900 E=AN
910 GOSUB 1100
920 FOR Q=0 TO E
930 IF KHAR(Q)>4 THEN 960
940 KHAR(Q)=1
950 GOTO 970
960 KHAR(Q)=KHAR(Q)+1
970 NEXT Q
980 PL=(PL=0)
990 GOTO 530
1000 CALL COLOR(9,1,5)
1010 CALL COLOR(14,1,7)
1020 FOR I=1 TO 8
1030 CALL COLOR(I,PC(PL),2)
1040 NEXT I
1050 CALL COLOR(10,PC(PL),1
6)
1060 CALL COLOR(11,PC(PL),1
1)
1070 CALL COLOR(12,PC(PL),0
)
1080 CALL COLOR(13,PC(PL),1
4)
1090 RETURN
1100 IF ((AN*(PS)+PL=0)+(A
N*(PS)+PL=1) THEN 1170
1110 A=(KHAR(PS)=1)+(KHAR(P
S)=2)*2-(KHAR(PS)=3)-(
KHAR(PS)=4)*2
1120 BP=BP+A*3
1130 PS=PS+A
1140 IF (PS>0)+(PS<10) THEN
1170
1150 PS=(PS=-1)+(PS=11)+PS

```

```

1160 BP=(BP<5)*3-(BP>29)*3
1
1170 RETURN
1180 R=14
1190 C=7
1200 A$=P$(-(PS=10))&" IS T
HE WINNER!"
1210 GOSUB 110
1220 A$="LIKE TO PLAY AGAIN
(Y/N)?"
1230 R=24
1240 C=4
1250 GOSUB 110
1260 CALL KEY(0,K,H)
1270 IF H=0 THEN 1260
1280 IF (K<70)+(K>89) THEN
1260
1290 IF K=89 THEN 330

```



This version of "Tug-a-War" uses several of the Amiga's 4,096 different color shades.

Program 7: Tug-A-War For Amiga

Version by John Krause, Assistant Technical Editor

```

SAY TRANSLATE$( "")
SCREEN 2,320,200,5,1-
WINDOW 2,"Tug-A-War",12,2
FOR I=0 TO 7-
  READ r,g,b-
  PALETTE I,r,g,b-
NEXT-
RANDOMIZE TIMER-
DIM A(11)-
FOR I=1 TO 11-
  A(I)=INT(RND(1)*4)+4-
NEXT-
row=3-
col=5:col=4:GOSUB frame:GOSUB squ
are-
col=4:col=5:GOSUB frame:GOSUB squ
are-
col=5:col=6:GOSUB frame:GOSUB squ
are-
col=6:col=7:GOSUB frame:GOSUB squ
are-
row=10-
LOCATE 5,11:COLOR 1,4:PRINT "2"-
LOCATE 5,14:COLOR 1,8:PRINT "1"-
LOCATE 5,25:COLOR 1,5:PRINT "1"-
LOCATE 5,29:COLOR 1,7:PRINT "2"-
LINE (64,5)-STEP(18,0),1-
LINE -STEP(8,4),1-
LINE (48,5)-STEP(6,-4),1-
LINE (24,36)-STEP(18,0),1-
LINE -STEP(-8,4),1-
LINE (384,36)-STEP(-8,-4),1-
FOR col=3 TO 18-
  GOSUB frame-
NEXT-
col=0:col=3:GOSUB square-
col=12:col=2:GOSUB square-

```

```

dot=5:GOSUB update-
SAY TRANSLATE$( "wlooms to tug o wa
r ")
main-
LOCATE 17,18-
IF red THEN-
  COLOR 2,0:PRINT "Red's turn "-
  SAY TRANSLATE$( "reds turn.")-
ELSE-
  COLOR 3,0:PRINT "Blus's turn "-
  SAY TRANSLATE$( "blues turn.")-
END IF-
WHILE MOUSE(0)<1 OR MOUSE(4)<3
0 OR MOUSE(4)>104 OR MOUSE(5)<2
5 OR MOUSE(5)>275-
WEND-
click=INT(MOUSE(3)/84)-
IF (red AND click<=dot) OR (red=0 AND
D click>=dot) THEN -
  temp=dot-
  IF a(temp)=4 THEN dot=dot-2-
  IF a(temp)=5 THEN dot=dot-1-
  IF a(temp)=6 THEN dot=dot+1-
  IF a(temp)=7 THEN dot=dot+2-
END IF-
IF red THEN-
  FOR i=click TO 11-
    a(i)=a(i)+1-
    IF a(i)=8 THEN a(i)=4-
    NEXT-
  ELSE-
    FOR i=1 TO click-
      a(i)=a(i)+1-
      IF a(i)=8 THEN a(i)=4-
      NEXT-
    END IF-
  IF dot>11 THEN-
    dot=12:GOSUB update-
    LOCATE 17,18:COLOR 2,0:PRINT " R
d wins! "-
    SAY TRANSLATE$( "red wins.")-
    GOTO quit-
  END IF-
  IF dot<1 THEN-
    dot=0:GOSUB update-
    LOCATE 17,18:COLOR 3,0:PRINT "Blu
e wins! "-
    SAY TRANSLATE$( "blue wins.")-
    GOTO quit-
  END IF-
  GOSUB update-
  red=1:red-
  GOTO main-
frame=-
x=24:IF 24*col>280 THEN x=23-
LINE (24*col,8*row)-STEP(x,24),1,b-
RETURN-
square=-
x=28:IF 24*col+1>280 THEN x=21-
LINE (24*col+1,8*row+1)-STEP(x,24),0
olr,bf-
RETURN-
update=-
FOR col=1 TO 11-
  col=5(col):GOSUB square-
NEXT-
CIRCLE (24*dot+11,91),5,1-
PAINT (24*dot+11,91),1-
RETURN -
quit=-
LOCATE 19,7:COLOR 1,0:PRINT "Clie
k mouse to play again."-
SAY TRANSLATE$( "click mouse to pla
y again.")-
WHILE MOUSE(0)=0:WEND-
RUN-
DATA 5,5,5,0,0,1,0,0,0,0,1,0,1,0,1,1,
0,1,0,1,0,1,-

```


If the sub is surfaced, you can climb to the bridge. From here you can look around to spot the enemy ships, using your naked eyes or binoculars. The view includes the Japanese ships (several varieties, each graphically different) and the land on the horizon. The landforms correspond exactly to where you are in the Pacific; if you patrol to the coast of New Guinea, you will see the coast of New Guinea. Given the amount of territory covered in the game, the mapping system is obviously very sophisticated.

The binoculars and periscope screens include all the details necessary for firing at the ships: target type, target range and speed, and such data as angle-on-bow. You may fire the deck guns or torpedoes, but only torpedoes are consistently effective.

These screens are the heart of the action, but to keep the sub running you must pay strict attention to your vessel's instruments and gauges. They display information about the battery, the depth of the sub and of the ocean floor beneath you, fuel levels, the status of hull openings, and so on. For instance, the battery allows restricted underwater maneuvering, depending on your speed, and then must be recharged on the surface.



This assortment of screens from *Silent Service* shows the diversity of the program's graphics.

Attention To Detail

Other facets of the game are equally realistic. If your sub hits the ocean bottom, you hear a scraping sound and the hull may be damaged. You can cruise at four speeds or cut the engines for silent running. More esoterically, once per mission you can get rid of your emergency tanks to stop a fatal dive, or release debris to the surface to fool the Japanese destroyers into thinking you've been sunk. Sound effects range from sonar pings—telling you that destroyers are closing in—to the ominous explosions of nearby depth charges. The sub's hull even creaks if you dive deeper than it was tested for, and you hear a grinding metallic sound if you're

rammed by an enemy ship. *Silent Service's* detail is astonishing.

But the most impressive part of the detail is that it does not impede play. Detail in the more traditional type of war game frequently hinders understanding and lengthens the game considerably, but *Silent Service* plays quite easily after only a half-hour or so of practice. Once you learn how to steer the sub and fire torpedoes at a target, you can try a mission. You can learn the rest, such as diving and running silent at the approach of a Japanese destroyer, as the situation demands. At any time, you can pause the game to allow you time to think.

Like *F-15 Strike Eagle*, *Silent Service* is both intriguing and addicting. Also like *F-15*, it is highly educational, but there is nothing tedious about the lessons. The excellent manual describes the submarine war in the Pacific, the background to the scenarios, and the tactics used by sub captains. By playing the game, you'll quickly find that these captains knew what they were doing. *Silent Service* is a superior product.

Silent Service
MicroPro Software Inc.
120 Lakefront Drive
Hunt Valley, MD 21030
\$34.95-\$39.95 (depending on version)

DeluxePaint For Amiga

Lee Noel, Assistant Editor, Art & Design

Requirements: Amiga with at least 256K RAM (512K recommended). Printer optional.

Whenever a new computer appears on the market, some of the most important factors affecting its success are the quality and diversity of its software. In the case of the long-awaited and innovative Amiga, questions concerning software support become even more important. Is this computer the powerhouse it's said to be, and can programs be written to take full advantage of its capabilities?

Electronic Arts, a software publisher widely considered to be at the forefront of personal computing, said yes to both questions and threw its considerable weight squarely behind the Amiga. *DeluxePaint*, by Dan Silva, is one of the first results. Not surprisingly—considering the Amiga's selling point as a computer for those who want a "creative edge"—*DeluxePaint* is a visual arts program of immense scope and flexibility.

In fact, *DeluxePaint* is really three different programs of immense scope

and flexibility. Due to differing memory requirements, *DeluxePaint* includes a separate program for each of the Amiga's three major screen modes: 320 pixels across by 200 down with 32 simultaneous colors; 640 × 200 with 16 colors; and 640 × 400 with 16 colors. The number of simultaneous colors in each mode can be selected from a palette of 4,096 possible colors. You can also customize *DeluxePaint* by restricting it to a smaller palette.

After booting up the program disk, you must type in a command to call up whatever incarnation of *DeluxePaint* you want. This may sound confusing to nontechnical artists hoping to use the Amiga for their first experiments in computer graphics, but loading the program is fairly straightforward. First turn on the Amiga and insert the usual Kickstart disk. When the prompt asking for the Workbench disk appears, you insert the *DeluxePaint* disk instead. AmigaDOS comes up next with its 1> prompt, and then you type the appropriate command. For instance, you'd enter *dpaint* and press RETURN to work in the 320 × 200 mode.

Best For 320 × 200

DeluxePaint works best by far in the 320 × 200 mode. In the 640 × 200 mode, pixels are three times as high as they are wide, and the program slows down considerably. In 640 × 400, the slowdown is drastic. So much memory is consumed that there's not enough room in a 512K machine for both the entire program and a screen. Instead, the program is broken into modules that are constantly swapped in and out from disk. (*DeluxePaint* works this way in all modes on a 256K Amiga.)

Also, the 640 × 400 mode suffers from a jittering screen display. The jittering varies depending on the color combinations, and high-contrast combinations are worse. This isn't *DeluxePaint's* fault—the monitor simply cannot refresh the 256,000 pixels in this mode fast enough to display a stable picture. (Other computers with similar modes get around this problem by using special monochrome monitors driven at higher refresh rates.)

Since *DeluxePaint's* features are the same in all modes, we'll describe what's available in the 320 × 200 mode. This is the most color-rich screen, and the program's documentation and all of the sample pictures on the disk are slanted toward it.

kyan

ALL NEW SOFTWARE LINEUP!

kyan pascal (Version 2.0) \$69.95

kyan pascal is the ideal system for learning Pascal and developing Pascal programs. It's a full implementation of ISO Pascal and conforms to the standards set by the Federal Software Testing Center. **kyan pascal** features a menu-driven environment with multiple HELP screens; a full-screen text editor; and, optimized 6502 machine code compiler/assembler. It produces code that runs at the maximum speed possible on the 6502 microprocessor. **kyan pascal** supports many extensions including string handling, linking, chaining, random files, and included or inline assembly source code. It also supports a line of powerful toolkits which make it possible for even novice programmers to develop sophisticated software. **kyan pascal** (Version 2.0) requires only one disk drive. It is available for the Apple II (runs in ProDOS and requires 64K); Atari (runs DOS 2.5 and requires 48K); and Commodore 64/128. **kyan pascal** is not copy protected and comes with a Pascal tutorial and reference guide.

Programming Utility Toolkit \$49.95

Programming is faster and easier with this extensive library of utility programs and file management procedures. The Toolkit includes source code for more than 20 utility programs.

Advanced Graphics Toolkit \$49.95

Add stunning graphics to your Pascal or assembly language programs. With the Toolkit's graphics primitives, you can build a custom graphics library. Or, you can use the Toolkit's library for 2 and 3 dimensional transformations, windows and clipping, shading, and more.

kyan Software offers you a 15 day money back guarantee. See for yourself... kyan is the best programming software.

Send Check/Money Order: kyan software, Dept. P • 1850 Union Street, #183 • San Francisco, CA 94123

Or Call: (415) 626-2080 • Visa/MC Accepted

Please include \$4.50 (order for shipping) handling, \$10 outside North America. CA residents add 6.5% sales tax.

MouseText Toolkit (available for Apple II only) ... \$49.95

Add Macintosh-like graphics to your Pascal programs. The Toolkit includes routines for pull-down menus, windows, and mouse-controlled cursor events (Toolkit requires Apple IIc or enhanced IIe).

Macro Assembler/Linker \$69.95

kyan's latest programming tool adds a new dimension to assembly language programming. The Assembler/Linker includes a text editor, 65C02 macro assembler, object module linker, debugger, and librarian.

Tons Of Tools

Like virtually all Amiga software, **DeluxePaint** is a mouse-driven, icon-based program, similar in some ways to **MacPaint** for the Macintosh. An array of drawing tools is represented by icons on the computer's display (the tools can be hidden when the picture is finished). Tools are selected by moving a pointer to the appropriate icon with the mouse. A click of the mouse button activates the tool, which can then be used in the drawing area.

DeluxePaint has practically all of the tools that have become standard in graphics-design programs. You can draw straight lines and a multiplicity of outlined or filled shapes, paint with different-sized brushes or an "air-brush," print text on the screen, and lots more. But **DeluxePaint** really shines because it offers unique new tools and novel extensions to the old standbys.

First and foremost, **DeluxePaint** recognizes the importance of color to the artist. The program has a special palette window where any of the Amiga's thousands of colors can be mixed and used in design work. The palette appears from the program's title bar as a pull-down menu (the menu selections have alternate keyboard commands as well.)



This picture of a paint can and brush— included on the DeluxePaint disk— shows the fine shading possible with 32 simultaneous colors chosen from a palette of 4,096.

Within the palette is an almost bewildering number of options. For instance, there are two ways to make color adjustments. The first method has three slide controls—manipulated with the mouse—that alter the percentages of red, green, and blue in any color (these are the primary colors for a video display). Another set of three sliders allows changes to the hue, saturation, and value of any selected color. The latter system is much like the tint, color, and contrast controls on a normal color TV. As a result, novice users of **Deluxe-**

Paint may find this system reassuringly familiar.

If that's not enough flexibility, the artist can also move the palette window to any convenient location, and the relocation will be "remembered" for the rest of the current session.

Flowing Colors

Colors are selected simply by pointing and clicking with the mouse; selection is verified by a highlighted box. Color changes are instantly reflected in the palette window and in the picture. As a consequence, it's delightfully easy to adjust colors relative to each other. The program disk includes two good examples of the effects made possible by this precise control over a diverse palette. "KingTut" displays the hard, gleaming gold coffin mask of the Egyptian boy-king Tutankhamen. In contrast, "Venus" faithfully reproduces the soft, almost pearly hues of Botticelli's "Birth of Venus."

Some really amazing special effects are also built into **DeluxePaint's** palette, such as animated color cycling. This allows the artist to establish three sets of colors that will cycle through a certain range. Each range can be narrow or wide, can include harmonious or clashing colors, and can overlap the

COLOR MONITOR

Sale

**16
COLORS**

(Premium Quality)

- Built In Speaker & Audio
- For Video Recorders
- For Small Business Computers
- Apple - Commodore - Atari - Aplus 3000 -etc.
- One Year Free Immediate Replacement Warranty*



(Premium Quality)

- Beautiful Color Contrast
- High Resolution
- Sharp Clear Text
- Anti-Glare Screen
- 40 Columns x 24 Lines
- Front Panel Controls

List \$329⁰⁰

18 Columns

RGB
Super High
Resolution

13" Color Computer Monitor*

- *C64/Atari composite cable \$9.95
- *C128 RGB/Composite 80 column cable \$19.95.

Sale \$159.95*

Add \$14.50 Shipping

13" RGB & COMPOSITE COLOR MONITOR

Allows use of C-128 and C64 mode - composite and 80 column RGB mode. Must be used to get 80 columns in color with 80 column computers. Specially designed for use with the C128's special composite video output, plus green screen only option switch. (add \$14.50 shipping)

List \$399.00

Sale \$259.95*

12" MAGNAVOX (NAP) 80 COLUMN MONITOR

Super high resolution composite green screen monitor. 80 columns x 24 lines, easy to read, plus speaker for audio sound included. Fantastic value. Limited Quantities.

List \$129.00

Sale \$79.95*

Turn Your Monitor into a TV Set Without Moving Your Computer

Elegant TV Tuner with dual UHF/VHF selector switches goes between your computer and monitor. Includes mute, automatic fine tuning and computer-TV selector switches. Inputs included for 300 ohm, 75 ohm, and UHF. Can be used with cable TV and VCR's. Fantastic Value. Limited Quantities.

List \$129.95

Sale \$49.95*

15 Day Free Trial - 90 Day Immediate Replacement Warranty

• LOWEST PRICES • BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add \$10.00 for shipping, handling and insurance. Illinois residents please add 6 1/2% tax. Add \$20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA. APO-FFO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail. Prices & Availability subject to change without notice. VISA — MASTER CARD — C.O.D. No C.O.D. in Canada. APO-FFO

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order

240K Apple® Compatible Computer System

NEW

SALE

Aplus 3000 computer system includes 192K RAM, 48K ROM (32K Microsoft Basic plus 16K ROM Emulator), 160K Laser 5 1/4" Disk Drive (Runs Apple II Software), Magic Window Wordprocessor, MagiCalc spreadsheet, Magic Memory Database. All for only \$399.00

Complete System

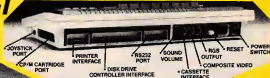
\$399.00

• **15 Day Free Trial**

"Aplus 3000" System



Runs Apple II Software



Double Immediate Replacement Warranty

If any of the Aplus 3000 computer system equipment fails due to faulty workmanship or material within 180 days of purchase we will REPLACE it immediately with no service charge!!

- Over 10,000 existing Apple® programs
- Centronics printer interface included
- 240K (192K RAM, 48K ROM)
- ArtSci's Magic Window II, Magic Memory, and MagiCalc included
- 160K Laser 5 1/4" Disk Drive (Runs Apple II software)
- RGB (80 columns in color) and composite included

SPECIFICATIONS

A plus 3000 is a complete, self-contained computer based on the popular 6502A microprocessor and can tap into the tremendous software library of Apple II. Features include 192K Bytes RAM, 32KB Enhanced Microsoft BASIC, 80 column text, 560H X 192V color graphic display, 81 key sculptured keyboard and high efficiency switching power supply. Also included as standard are Centronics bus printer interface, Cassette interface, 4 channel sound generator, and 5 1/4" Apple Compatible Disk Drive.

• TEXT

- 40 columns X 24 rows or 80 columns X 24 rows software selectable.
- 5 X 7 characters in 7 X 8 matrix.
- Upper and lower case characters.
- One of Eight colors for characters/graphics and background, Red, Green, Blue, Cyan, Magenta, Yellow, Black and White.
- Character set with normal, inverse and flashing capabilities.

• GRAPHICS

- 260H X 192V 6 colors — Black, White, Violet, Green, Blue, Orange.
- 260H X 192V 8 colors bit image — Black, White, Red, Green, Blue, Cyan, Magenta, Yellow.
- 560H X 192V 6 colors — Black, White, Violet, Green, Blue, Orange. (High resolution color monitor required)

Super Apple Compatible Disk Drive Sale \$149.95. Quicker, Cooler, Better Disk Drives for your Apple II plus, IIe, IIc (specify when ordering). List \$289.95. Sale \$149.95.

More Features than Apple® for less than Commodore®

Features	Aplus 3000	Apple IIe	Commodore C-128
RAM	192K	64K	128K
Runs Apple II Software	Yes	Yes	No
Function Keys	24	None	16
4 Voice, 6 Octave Sound	Yes	No	Yes
Composite Video	Yes	Yes	Yes
Disk Drive	Included	Extra Cost	Extra Cost
Numeric Keypad	Included	Extra Cost	Included
Video Cable	Included	Extra Cost	Extra Cost
RGB Color Card	Included	Extra Cost	Included
80 Column Card	Included	Extra Cost	Included
Centronics Printer Interface	Included	Extra Cost	Extra Cost
Drive Controller	Included	Extra Cost	Included
\$150 Wordprocessor (Magic Window)	Included	Extra Cost	Extra Cost
\$150 Spreadsheet (MagiCalc)	Included	Extra Cost	Extra Cost
\$60 Database prg. (Magic Memory)	Included	Extra Cost	Extra Cost
Your Cost	\$399.00	\$1745.00	\$1117.90

ACCESSORIES

	LIST	SALE
2nd Disk Drive	\$299.95	\$149.95
2 professional analog joysticks	\$ 39.95	\$ 24.95
Z-80 cart. allows CP/M use	\$ 99.95	\$ 59.95
RS232C adapter	\$ 99.95	\$ 59.95
R/F Modulator (TV hookup)	\$ 29.95	\$ 19.95
RGB cable (RGB Monitor hookup)	\$ 24.95	\$ 19.95
Centronics cable (for Centronics printer)	\$ 34.95	\$ 24.95
Technical reference manual	\$ 29.95	\$ 19.95
80 column Hi-Res Green Monitor	\$199.00	\$ 79.95
80 column Hi-Res RGB Monitor	\$399.00	\$259.00

Add \$25.00 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail!! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add \$10 more if C.O.D.

15 Day Free Trial — If it doesn't meet your expectations within 15 days of receipt, just send it back to us UPS prepaid and we will refund your purchase price!!

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order

Famous Smith Corona National Brand

10" PRINTER SALE

Below Wholesale Cost Prices!!!

*** ONE YEAR IMMEDIATE REPLACEMENT WARRANTY**

- Speed: 120 or 160 characters per second
- Friction Feed/Tractor Feed — Standard
- 80 character print line at 10 CPI
- 1 Line Buffer, 2K Buffer on 160 CPS Plus LQM
- Six pitches
- Graphics capability
- Centronics compatible parallel interface
- Features Bidirectional Print, Shortline Seek, Vertical And Horizontal Tabs



SUPER GRAPHICS

This is a sample of our near-letter-quality print. *emphasized italic print.* There is standard data processing quality print

Check These Features & Prices

120 CPS 10" Printer

List \$429.00
SALE \$159

160 CPS + Letter Quality Mode 10" Printer

List \$499.00
SALE \$199

(IBM — Commodore)

SPECIFICATIONS

(Apple — Atari — Etc.)

Size/Weight

Height 5.04" Width 16.7"
Depth 13.4" Weight 18.7 lbs.
Internal Char. Coding
ASCII Plus ISO

Print Buffer Size

120 CPS: 132 Bytes (1 line)
120/160 CPS Plus LQM: 2K
No. of Char. in Char. Set
96 ASCII Plus International

Graphics Capability

Standard 60, 72, 120 DPI
Horizontal 72 DPI Vertical
Pitch

Pitch

10, 12, 16.7, 5, 6, 8.3, Proportional Spacing

Printing Method

Impact Dot Matrix

Char. Matrix Size

9H x 9V (Standard) to 10H x 9V
(Emphasized & Elongate)

Printing Features

Bi-directional, Short line seeking, Vertical
Tabs, Horizontal Tabs

Forms Type

Fanfold, Cut Sheet, Roll (optional)

Max Paper Width

11"

Feeding Method

Friction Feed Std., Tractor Feed Std.

Ribbon

Cassette — Fabric inked ribbon

Ribbon Life

4 million characters

Interfaces

Parallel 8 bit Centronics compatible
120/160 CPS Plus NLQ: RS232 Serial Inc.

Character Mode

10 x 8 Emphasized: 9 x 8 Standard, 10 x 8
Elongated: 9 x 8 Super/Sub Script (1 pass)

Character Set

96 ASCII
11 x 7 International Char.

Line Spacing

6/8/12/24 LPI

Character Spacing

10 cpi normal; 5 cpi elongated normal; 12 cpi
compressed; 6 cpi elongated compressed;
16.7 cpi condensed; 8.3 cpi elongated
condensed; 5.12.5 cpi elongated proportional

Cartridge Ribbon — List \$19.95. Sale \$12.95.

Interfaces

IBM \$89.00

Apple \$59.00

Atari \$59.00

Commodore \$39.95

Add \$14.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add \$29.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA. Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days delivery. 2 to 7 days for phone orders. 1 day express mail! Prices & Availability subject to change without notice.
VISA — MASTERCARD — C.O.D. No C.O.D. to Canada or APO-FPO

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order



152K *Lowest Price In The USA!* 152K

Computer System Sale

• Students • Word Processing • Home • Business

152K System **\$379***
(130XE System)

EDUCATE WITH ATARI



LOOK AT ALL YOU GET FOR ONLY **\$379**
LIMITED QUANTITIES

\$379
SYSTEM PRICE

LIST PRICE

INDIVIDUAL
SALE PRICE

SAVE
OVER \$100
ALL 5 ONLY
\$379.00
SYSTEM
SALE PRICE

① Atari 130XE 152K Computer

② Atari 1050 127K Disk Drive

③ Atari 1027 Letter Quality 20 CPS Printer

Atari Writer Plus Word Processor with Spell Checker

Atari BASIC Tutorial Manual

All connecting cables & T.V. interface included.
* Monitors sold separately.

\$249.00

299.00

299.00

59.95

16.95

\$134**

159**

159**

49**

12**

TOTALS

\$923.90

\$517.75

CALL FOR 1027 PRINTER REPLACEMENT OPTIONS

Other Accessories

- ☆ 12" Hi Resolution Green Screen Monitor
- ☆ 13" Hi Resolution Color Monitor

List

Sale

\$199.00

\$399.00

\$79.95

\$159.95

Add \$9.95 for
Connection Cables
Add \$10 for UPS

15 DAY FREE TRIAL. We give you 15 days to try out this ATARI COMPUTER SYSTEM! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!! **90 DAY IMMEDIATE REPLACEMENT WARRANTY.** If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service
• One Day Express Mail • Programming Knowledge • Technical Support

Add \$25.00 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery - 2 to 7 days for phone orders - 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add \$10 more if C.O.D., add \$25 if Air Mail.

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order

ranges for the other two cycles. Once activated, a cycle runs through all the colors in its range in a smooth sequence. The speed of each cycle can be individually controlled with the mouse and a slider. Color cycling is what makes the disk's sample waterfall picture seem to flow. With thought and care, you can create effects otherwise impossible in two-dimensional art.

Closer to traditional art media are tools which smear, shade, and blend. Smearing enables an artist to use the *DeluxePaint* brush to smudge colors already painted on the display. The effect is similar to running a brush through fresh oil paints. Shading and blending work on a defined color range, like the ranges for color cycling. Both operate best on a range of closely related shades and affect only the colors in the selected range.

The action of shading is difficult to explain in print, but blending does pretty much what it describes—it produces smooth gradations like those found in watercolor washes or airbrushing. There are also tools for exchanging colors and an undo feature to recover from mistakes.

Anything Is A Brush

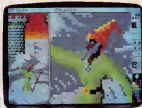
All that's fine for the colors, but what of the brushes that apply them? Once again, *DeluxePaint* offers abundant options. There are, of course, some built-in brushes. These are various useful shapes, and they can all be adjusted in size with one of the program's easy-to-use tools. But, as might be expected, other brush manipulations range into some wild and unexplored regions.

The essence of *DeluxePaint* brushes is this: Anything can be a brush, so you can paint with a single pixel, a pictorial element, a whole screen, or anything in-between. And if that's not enough, the artist can grab the brush, resize it, rotate it to any angle, flip it, or drag it into a completely new shape. This last feature allows a brush that looks flat to be made to appear three-dimensional. To get an idea of the effect, imagine a flag lying flat on a table. Suddenly, it's bent into a stiff billow and paints in an arc across the sky—stars and stripes and colors and all.

Brushes, like complete pictures, can be saved to previously prepared data disks. There are also some esoteric possibilities involving the exchange of transparent and solid colors within a brush, plus a different way to "hold" the current brush, and a way to speed up response time to certain types of brushes. Analogies are hard to come by for these features, but they open up fascinating possibilities not available to



Another sample picture on the *DeluxePaint* disk is this reproduction of Botticelli's "Birth of Venus."



This photo demonstrates a zoom window, just one of the many features in *DeluxePaint*.

artists working in traditional media.

The basis for all the brush transformations is the designer's ability to reach out and capture any area of the screen with a special brush selection tool. This is much like the copy, cut, and paste functions found in other graphics programs, and it can be used for those purposes as well as brush design.

Room To Zoom

A full description of *DeluxePaint*'s myriad features would run on for many pages (the manual is 31 pages long), but highlights of some of the major ones not covered so far bear mentioning.

DeluxePaint has a special magnification tool that allows the artist to zoom closer and closer to the area under inspection, and then back away in similar increments. Great mobility within the magnify mode is provided with the cursor keys.

There's fairly complete printer support, plus the ability to add text to designs and manipulate it in numerous ways. A skewing feature even lets you turn ordinary text into italics.

For precision design work, one option gives a constantly updated display of the cursor's screen coordinates, another provides a grid that can be modified, and still another allows unusual mirror-like symmetry effects.

Virtually anything that anyone ever wanted in a personal computer graphics program is included in *DeluxePaint*—and it's all easy to use and easy to learn. It's fortunate that the program is fairly intuitive because the documentation is not. The basic tone of the manual is that experimentation and playfulness are the best methods for coming to grips with the program. A step-by-step approach might have been more helpful. And, ironically, the manual's graphics are almost nonexistent.

Another problem with the manual is that it lacks completeness. *DeluxePaint* cannot create data disks by itself, so the artist must refer to Chapter 4 of the *Amiga User's Guide* for the information. Also, hardly any program commands are summarized in *DeluxePaint*'s command summary.

On the other hand, in the few places where you might get really stuck, the documentation comes through with some solid tutorials.

Bottom line: Will *DeluxePaint* and an Amiga give you that creative edge? That depends—in the end, it's still the artist that has to pull the rabbit out of the hat.

DeluxePaint
Electronic Arts
2755 Campus Drive
San Mateo, CA 94403
\$79.95 (Introductory price)
\$99.95 (Regular price)

S'More For Commodore 64

Art Hunkins

Requirements: Commodore 64 or a Commodore 128 in 64 mode.

Commodore 64 owners who wish to upgrade their computers have two main options: Buy a Commodore 128 or install a *S'more* cartridge from Cardco. Each choice has its advantages. If money is no object (and you aren't overly attached to your 64), you might consider the 128. But the choice isn't that clear-cut. For those who write their own BASIC programs, *S'more* has some significant advantages of its own. Frankly, it's difficult to know which to compare *S'more* to—the 64 without *S'more*, or the 128.

Of course, the 128 does have some things going for it: twice the available user memory (122,365 bytes) as the 64; BASIC 7.0, with powerful commands for graphics, sprites, sound, and windowing; and a FAST mode for double-speed operation. So if it's raw computer

power and extra memory you want, the 128 is hard to beat.

On the other hand, *S'more BASIC* is more comprehensive than *BASIC 7.0* in its utilities; it defaults to disk *LOAD*, offers a greater variety of input options as well as more flexible screen formatting, and includes varied reset options. The built-in utilities are a real boon: *MERGE*, *AUTO*, *HEX*, *DEC*, *FIND*, *CHANGE*, *reNUMBER*, *DUMP*, and *OLD*—all familiar to *BASIC* *AD* users. The *LIST* command can scroll up and down, not true of *BASIC 7.0*. On the 128, only *AUTO*, *RENUMBER*, and a disk file *APPEND* are implemented.

Compared to the unenhanced 64, *S'more* frees up 57 percent more user memory—61,183 bytes instead of 38,911 bytes. The memory is contiguous and can be used in any way you desire. (As we'll see, there are other protected locations where machine language routines up to 512 bytes long may be stored.)

Improved Disk Commands

S'more BASIC and *BASIC 7.0* come out about even when it comes to disk commands (a notable weakness with the unexpanded 64); only the approach is different. Whereas 7.0 gives a wealth of specific commands, *S'more* uses only one—*DISK*, an all-purpose "wedge" followed by the traditional disk access symbols. Both *BASICs* also offer numerous enhancements of standard commands (such as a *RUN* that *LOADs* and *RUNs* a *BASIC* program from disk). Both permit the *SHIFT-RUN* key combination to *LOAD/RUN* the first program on disk.

Both *BASICs* offer about the same range of programming structures (*DO-LOOP*, *WHILE-UNTIL*, *IF-THEN-ELSE*). Both implement error-trapping and *HELP*, and both have programmable function keys, though 7.0 sets aside almost twice the buffer (246 bytes versus 128) for key definitions.

S'more is also handy in that its *LOAD* and *SAVE* commands default to disk (there is no *DLOAD* or *DSAVE*), and that it includes a disk *CATALOG*/directory option. In fact, due to the way the disk default option works, you can display the *CATALOG*, cursor to the program you want, type *LOAD* (or *RUN*), and hit *RETURN*—without worrying about what is displayed after the program name.

ML Limitations

For *BASIC* programs, *S'more* is superb. But let's look at *ML* applications. Here the picture is not so clear.

Although *S'more* has a *MONITOR* command, it doesn't have a built-in monitor; *MONITOR* just links you to a

monitor if you've loaded one into memory. *S'more* comes with a disk of software that includes a version of *Micromon* called *Smon*. (Other programs on the disk illustrate applications of the more noteworthy *S'more BASIC* extensions.)

Cardco's manual is thorough, clear, instructive, and particularly forthright when it describes *S'more's* limitations with memory addressing and machine language. Here's the catch: To make so much contiguous *BASIC* memory available, Cardco had to change a lot of memory locations and reconfigure memory. Cardco did what it could to maintain compatibility with *Commodore 64 BASIC (BASIC 2.0)*, but there were limits on what was possible.

It's remarkable that low memory with *S'more* is so highly compatible with *BASIC 2.0*. Only two differences will be noticed by the average programmer. First, and most importantly, the cassette buffer has been moved. *ML* programs designed to reside there will have to be transported to the new location. Also, some of the previously free bytes (which you may have used for flags or temporary data storage) are free no longer (zero page 251-254 remain available, however). There is a bonus, though—a 512-byte *RS-232* input/output buffer, protected from *BASIC*, which can be utilized for *ML* routines in most cases.

The most critical low memory locations for the *BASIC* programmer, the keyboard buffer and its corresponding character counter, remain intact. As the manual clearly states, however, *ML* routines that access *ROM* are in for major rewrites. The only *ROM* routines that are safe to use are the *Kernal* routines when they are accessed through the vectors in low memory (these vectors are unchanged in location). You cannot access *ROM* subroutines directly. This is a problem particularly with the *SID*, *VIC*, and *CIA* chips—that is, when working directly with screen, sound, and input/output peripherals.

The S'more Solution

To get around these limitations, the manual suggests that perhaps most *ML* routines are best written in *S'more BASIC*, then compiled with the (not-yet-released) *S'more BASIC* Compiler. This suggestion indicates the degree of potential difficulty in converting most *ML* programs for use with *S'more*.

But there's another alternative, too. *S'more* establishes a set of *CIA*, *VIC*, and *SID* reserved variables (*DIMENSIONed* arrays). Each variable corresponds to a *CIA*, *VIC*, or *SID* chip location you might wish to *PEEK* or *POKE*. To *POKE* the location, just assign the variable the

desired value; presto, the *POKE* is done. To *PEEK* the location, just use the reserved variable in an expression. It works fine and is simpler than actually *PEEKing* and *POKEing*. For sound and the *SID* chip, for example, it is not too far from the convenience of using *BASIC 7.0's* new sound commands (*PLAY*, *FILTER*, *ENVELOPE*, etc.).

Of course, this technique works only from *BASIC*, not machine language. There are times when, for speed and efficiency, *ML* is required. Although conversion of *ML* routines accessing the support chips is possible, it is apparently far from trivial. (The manual does not attempt to explain; it only hints that *RAM/ROM* bank-switching is involved, and that the banking system is similar to that of the *Commodore Plus/4*.)

There is but one other limitation I've noticed with *S'more*. When writing or editing a *BASIC* program, the enhanced *BASIC* often responds slowly, particularly with long programs. The cursor can take 1.5 to 2.5 seconds to reappear after you hit *RETURN* to enter a new line; it takes longer toward the beginning than at the end of a program. On the other hand, garbage collection purportedly is speeded up dramatically over 2.0 *BASIC*.

And your Earls and Viscounts. If you've got royal ancestors, we have the noble software that can help you trace them down.

Family Roots and your Apple, IBM, Commodore, Kaypro*, and many others, offer individual and group sheets, charts, name indices, general search and text capabilities. Adapts to most disk drives, printers, and screens. You get more utility programs, plus lots of personal control. A comprehensive (new) manual is included.

All for just \$185.

Write or call today for more information and a free brochure.

Quinsept, Inc.

P.O. Box 216
Lexington, MA 02173
(617) 641-2930

American Express,
Visa, and MasterCard
gladly accepted.

*Trademarks for Apple
Computer Inc., International
Business Machines, CIB, Inc.
and Digital Research



Works With 128, Too

These are the only problems I've experienced working with the *S'more* cartridge. Overall, *S'more* maintains a high degree of compatibility with BASIC 2.0 (and its associated memory configuration), offers more than 50 percent additional memory accessible to BASIC, and a greatly enhanced language. It makes working with the screen and sound a simpler task for BASIC programmers.

In short, *S'more* is a cost-effective alternative to a Commodore 128 upgrade. (Cardco's literature describing *S'more* as a "bridge to the 128" is on target.) And even if you do decide later to acquire a 128, *S'more* works identically on the 128 in 64 mode.

S'more
Cardco, Inc.
300 S. Topoka
Wichita, KS 67202
\$69.95

Heart Of Africa

Neil Randall

Requirements: Commodore 64 or 128 in 64 mode; Apple II-series computer with at least 64K RAM; or an Atari 400/800/XL/XE with at least 48K RAM. Disk only.

One of the truths in the entertainment industry is that anything popular will spawn many imitators. The field of computer games is certainly no exception. The first hits were *Space Invaders*, then *Pac-Man*, and then *Donkey Kong*. Each of these games begat a host of imitations, few of which approached the quality of the original. Imitations are rarely as good as the things they imitate.

Still, for every imitator trying to capitalize on the popularity of someone else's game, there's a designer trying to improve upon his own original design. This is not imitation—at least not what we normally think of as imitation. Instead, the designer is making an honest effort to improve on a game concept. Much like an artist, who chooses a medium and produces work after work trying to perfect his use of that medium, the game designer invents a system, then produces game after game to develop the system to its fullest. As long as the system keeps improving, the enterprise is justified.

Such is the case with Ozark Software's *Heart of Africa*, published by Electronic Arts. *Heart of Africa* is an extension of the system pioneered in Ozark's own *Seven Cities of Gold* (re-

viewed in the September 1984 issue of *COMPUTE!*). Far from an imitation, it improves on the original game and offers a fresh approach to a system that many people considered near-perfect already. Both games deal with exploration, but *Heart of Africa* gives us something more: a quest.

In Search Of A Tomb

Your quest in *Heart of Africa* is to find the lost tomb of Ankh Ankh, somewhere in the middle of the Dark Continent. You travel alone, buying supplies and tools wherever you can. As you cross the continent, you make discoveries and try to obtain clues about the lost tomb from tribal chiefs. It's not hard to get information, but it's very hard to get useful information, and just as hard to stay alive. The perils are constant, from dying of thirst in the Sahara Desert to suffering a fatal bite by a poisonous snake.

Like *Seven Cities of Gold*, *Heart of Africa* is entirely joystick-driven. You can put your feet up, lean back in your easy chair, and play the game without touching the keyboard. For further playability, the game offers a diary that continually updates itself. The diary is a graphically attractive series of pages that records special events. On the surface, it seems only a nice addition to the game, but in play it greatly eases record-keeping. Any exploration game, be it a text or graphics adventure, demands some keeping of records: map-making, recording conversations, jotting down clues. But *Heart of Africa* takes most of these out of your hands. The map is produced for you on the screen, and your observations, even conversations, are recorded in the diary. You can read the diary at any point simply by loading it from disk. It makes the game extremely playable, especially for those who loathe keeping records.

The *Heart of Africa* game screen shows a solitary figure marching across the map. As you walk, the map scrolls north, south, east, or west, shedding light on more and more of the Dark Continent. The map is constantly updated, and you can check it at any point during the game to see what you've already discovered. As you travel, you discover villages, mountain ranges, rivers, lakes, and, of course, if you work hard enough, the source of the Nile.

Random events are sometimes positive, such as finding valuable caches left behind by previous explorers, as well as negative, such as encounters with crocodiles, poisonous snakes, or rhinoceri. If you're equipped with the right weapons, you can normally stave off an attack, but you may become ill, fatigued, or very thirsty. You

can paddle a canoe along the rivers and lakes, and you can even go over waterfalls. The entire continent is yours to discover.

Tribal Relations

Perhaps the most impressive part of the game is the interaction with the tribes. As in *Seven Cities of Gold*, where cooperating with the natives established your reputation, working with the tribes in *Heart of Africa* is difficult. Each tribe is different and each chief reacts differently to you. For some tribes, a few gifts will yield helpful information. For others, all the gold in the world seems insufficient. You can steal supplies by wielding your gun, but your reputation will suffer. Or worse, you may catch a blow dart. The only way you can know how a tribe will react is to visit each village. If you do well and reward the chief, he'll tell you what else you might bring for more information. If you do poorly, you'll be drummed out of the village.

The *Heart of Africa* manual consists primarily of the notes written by your predecessor, the person sending you on this mission. It describes each of the areas of Africa and the tribes therein. An impressive document for its sheer information, it is also vital for gaining clues about where you should go. It gives, for instance, translations of the tribal names for geographical points. To the natives, after all, Victoria Falls is not Victoria Falls.

There is nothing easy about the game, but the difficulty comes from the situation, not in trying to learn the system. It is extremely easy to get across Africa, buying things, finding things, and giving things away, but it is very hard to gain useful information. Still, this is the game's strength. A poor game is difficult to learn and offers few rewards. A good game is easy to learn and offers endless rewards. *Heart of Africa*, in this sense, is a very good game.

Like *Seven Cities of Gold*, *Heart of Africa* is professional in every way. An excellent program, filled with surprises, the game is even more addicting than its predecessor. In *Seven Cities of Gold*, your rewards were the excitement of discovery and the favors of your monarch. *Heart of Africa* duplicates the excitement of discovery, but adds a desperate search for a lost tomb. This quest makes *Heart of Africa* an adventure as well as a simulation.

Only One World To Explore

One of the superb features of *Seven Cities of Gold* was its ability to create new worlds to explore. Players could never exhaust the game because the

Lycos Computer Marketing & Consultants



1091....\$228

SEIKOSHA
SP-100UC (C-64) 169
SP-100CA Centronics 185

C. ITOH
Printer 3510 sp+ Call
15505 sp+ Call
Protobridge Call

TOSHIBA
P-340 489
P321+ 129
P341P 990
P341S 990
SP-1000 (NEW) 990

SAVE ON THESE IN STOCK PRINTERS

CARDCO
32K BUFFER (C-64) 59

CORONA
LP265 Laser Printer 2995
200361 Toner Cartridge 89

EPSON
LX80 209
FX85 353
FX80 193
HomePrinter 10 163
DX20 267
DX3 67
AP-30 244
H-80 350
H3-50 198
FX-286 (NEW) 489
LQ-800 (NEW) 529

CITIZEN

MSP-10 255
MSP-15 255
MSP-20 277
MSP-25 495

OKIDATA

Okidata 10 179
Okidata 20 207
182 214
185 214
32K BUFFER 89
32K BUFFER 135
64K BUFFER 645

BROTHER

Q25 549
430 API 1599
430 ECS 2759
P 30 CCI 699
P 3 119
P 113 879
DL 25 (NEW) 499

JUKI

Juk. 8160 844
R5333 Serial Board 66
P196 T-tactor 119
P196 Serial Printer 209
Juk. 6366 757

LEGEND

880 168
1080 291
1380 229
139 269
838 148

DIGITAL DEVICES

CAI 214
32K BUFFER 89
64K BUFFER 135

DIABLO

D25 549
430 API 1599
430 ECS 2759
P 30 CCI 699
P 3 119
P 113 879
DL 25 (NEW) 499

SG-10...\$205

PANASONIC
109 238
3131 (NEW) 264
3152 325
3161 404
1150 (NEW) 209
1582 (NEW) 439

SILVER REED

EXP400 299
EXP200 299
EXP400 299
EXP770 749

STAR MICRONICS

SG-10 205
SG-10C 219
LEWIP Basic 267
LEWIP Merge Print 319
SG-15 435
SG-15 435
SG-15 435
SG-15 435
SG-15 435

DUST COVERS

Atari
8205 11.95
1300X 8.99
8203AL 8.99
1053 8.99
1028 7.99

Commodore
C128 7.99
16T1154 5.99
1602 12.95
1702 8.99
OS4VLC 6.99

Panasonic
1902/1991 6.99
1292 6.99
1293 9.99

Star Micronics
SG502 8.99
SG503 8.99
SG10 5.99
SG15 8.99

Okidata
8250 8.99
8255 9.99
150 9.99

DRIVES

INDUS
Indus GT Atari 199
GT Commodore 209

MSD
301 Drive 229
301 Drive 229

TANDON
320K 1/4" 115

TYMAC
100-645 1/4" Apple 209

PRINTER INTERFACING

AXIOM
AT818 (Atari) 45
Parvix (C-64) 45

MICROBITS
MPP-1150 (Apple) 54
MPP-150K (Apple) 54
MicroPrint (Atari) 39
64K RAM Board (8008) 55

DIGITAL DEVICES
AppleFace XLP (Atari) 49
U-Print A (Atari) 49
U-Print C (C-64) 49
U-Print A18K Buffer 74
U-Print A18K Buffer 99
U-Print A18K Buffer 99
U-Print A18K Buffer 139
U-CALL RS232 (Atari) 37

MICROTEK
Dumping GS (Apple) 59
Dumping 16K (Apple) 49
RV-611C (Apple) 49

TYMAC
Connection (C-64) 55
Jackler (Apple) 49
MPC-100 (Apple) 39

ORANGE MICRO
GRAPPLER+ (Apple) 85
Grappier 16K (Apple) 149
GRAPPLER (Apple) 69
Grappier CD (C-64) 79

CARDCO
G-Wire (C-64) 54
Crv-a (C-64) 45
Crv-b (C-64) 49
Crv-c (C-64) 49
Crv-d (Atari) 49
Crv-e (Apple) 49

DISKETTES

GENNISON
ELEPHANT 5 1/4" 5DD 11.99
ELEPHANT 5 1/4" 5DD 12.99
ELEPHANT 5 1/4" 5DD 13.99
PREMIUM 5 1/4" 5DD 13.99

SUNKYONG
5 1/4" 5DD 11.99
5 1/4" 5DD 13.99

MAXELL
5 1/4" MD1 13.99

VERBATIM
5 1/4" 5DD 13.99
5 1/4" 5DD 13.99

BONUS
5 1/4" 5DD 8.99
5 1/4" 5DD 12.99

NO LABEL DISKETTES
NL 5 1/4" 5DD 15.99 (Box 10)
NL 5 1/4" 5DD 15.99 (Box 10)
Free Diskette Writor Pen!
Free Storage Case!

IBM-PC SOFTWARE

***LOTUS**
Lotus 1.23 309.00
Symphony 439.00

***QUADRAM**
Quad Jr. Exp. Classes 619.00
Quad Jr. Exp. Memory 209.00
Quad Jr. Exp. Memory 209.00
ITT XTRA XP Personal Computer 299.00
256K 2 Drive System CALL
256K 10 Meg Hard System Call

IBM-PC COMPATIBLE

***LEADING EDGE**
MetaSoft 69.95
LEWIP Basic 65.00
LEWIP Merge Print 65.00
LE Spell Correction 169.00

SSI (IBM)
Baffle for Summary 24.95
Baffle for Summary 24.95
Tegris in Snow 24.95
LEWIP Basic 24.95
LEWIP Merge Print 24.95
Ward Plotter 24.95
Ward Plotter 24.95

SYNAPSE (IBM)
Synapse 54.95
Ease 26.95
Wizard of Wall St. 26.95
Business 26.95

SUB LOGIC (IBM)
Jet Simulator 34.95

BRODERBUND (IBM)
Bank & Writer 34.95
The Print Shop 34.95
Graphics Library I 22.95
Look Point 22.95
Ancient Art of War 22.95
Chump Luck Runner 22.95

MICROPROSE (IBM)
+10 Stone Eagle 29.95
Soko Flight 29.95
Hollow Ace 29.95
Sphinx Ace 29.95

AT&T
Saturn 7300 CALL
6300 CALL

TOLL FREE 1-800-233-8760



TO ORDER



CALL TOLL FREE 1-800-233-8760
In PA 717-484-1030
Customer Service 717-494-1870

or send order to
Lycos Computer
P.O. Box 5083
Jersey Shore, PA
17740

RISK FREE POLICY

Include items shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. (excludes Alaska and Hawaii). PA residents add sales tax. APO, FPO, and international orders add \$3.00 plus 2% for money order service. Advertisers must show 4% discount for cash and 4% for MasterCard or Visa. Personal checks require a wire transfer before shipping. Ask about UPS Blue and Red label shipping. All merchandise returned under manufacturer's warranty. Free catalog with order. All items subject to change without notice.

Lyco Computer Marketing & Consultants



ATARI

800XL	CALL
1300XE (NEW)	28.95
1300XE (NEW)	28.95
1050 Drive	18.95
1010 Recorder	42.95
1050 Printer	25.95
1025 Printer	15.95
1087 Printer	19.95
850 Interface	15.95

ATARI

VIP Professional	75.00
ST Companion	15.95

ATARI SOFTWARE (NEW)

Codemaster	26.75
Pileover	26.75
Reporter	26.75
MenuMaster	26.75
Home Integrator	19.75
Small Bus Inventory	11.75
Salesman Expenses	11.75
Accs. Reentry	11.75
Retail Invoice	11.75
Print Legacy	15.75
Adventure Winner	19.75

BRODERBUND (Atari)

The Print Shop	28.95
Graphics Library	18.95
Graphics Library II	18.95
Whistler's Brother	18.95
Scenewriter	18.95
Sketch	18.95

INFOCOM (520ST)

Wambler	26.95
Delta Drawing Room	19.95
Cosmic Life Room	19.95
Up for Grabs Room	19.95
Word	31.95
Word II	31.95
Leadline	33.95
Sources	33.95
Suspect	31.95
Suspended	35.95
Culturobs	26.95
Star Straker	26.95
Wilde	31.95
Sorcerer	31.95
Champion	26.95
Witness	26.95
Planetoid	26.95
Spellbreaker	33.95

SPINNAKER

Delta Drawing Room	19.95
Cosmic Life Room	19.95
Up for Grabs Room	19.95

PERSONAL PERIPHERALS

Super Switch C-64	29.95
Printer Utility C-64	14.95
Super Switch II Atari	55.00
Super Switch Apple/II	55.00

MICROPROSE (Atari)

Kennedy Approach	21.75
Crusade in Europe	21.75
Decision in Desert	21.75
Solo Flight	30.75
Nato Commander	20.75
F-15 Strike Eagle	30.75
Helical Ace	18.75
Acropolis	21.75
Secret Service	21.75
Gunsling	21.75

SUB LOGIC (Atari)

Fight Simulator II	32.75
Night Mission Pinball	20.75

GLENTOP

ST Logo Book	15.95
--------------	-------

MARK OF UNICORN (520ST)

HEX	29.95
WHICE	129.95
PokerComm	99.95

FTL GAMES

Bundog	29.95
--------	-------

SIERRA

ULTIMA II	30.95
-----------	-------

HABA

10 Megabyte	CALL
Hippo C Compiler	49.95
Chess Master	49.95
Business Letters	39.95
Wills	39.95

SOFTTECHNICS

TimeLog	75.00
Rhythm	39.95

US ROBOTICS

Passport 12390	129
Passport 12391	129
Passport 20304	149
Passport 20307	139
Autodial 213A	319
PCMS	119
PCMS2	169
9100	379
Celebr	469
Modem	469
Tele-M-300	79

TELE MARKING

Telecommunications	29.95
(C-64)	29.95
Super 200 Super Apple	59.95
Super 200 Super IBM	59.95

SCARBOROUGH (Apple)

NET WORTH	49.75
Improved Mergotype	29.75
Mergotype Riser	29.75

BRODERBUND (Apple)

The Print Shop	31.95
Graphics Library	18.95
Graphics Library II	18.95
Paper Print	11.95
Bank St. Writer 64K	41.95
Bank St. Writer 128K	42.75
Bank St. Writer	42.75
Bank St. Speller	42.75
Carmen Sandiego	22.75
Kartlandia	22.75
Capitan Goodnight	22.75
Muppet Cruise	22.75

SUB LOGIC (Apple)

Fight Simulator II	32.75
Night Mission Pinball	20.75

MICROPROSE (Apple)

Crusade in Europe	21.75
Decision in Desert	21.75
F-15 Strike Eagle	30.75
Nato Commander	20.75
Acropolis	21.75
Secret Service	21.75
Gunsling	21.75

SSI (Apple)

Mech Brigade	38.75
Norway 1985	22.75
Mars/ Garden	29.75
St. Gun Shootout	24.75
50 Mission Crush	24.75
Phantoms	24.75

INNOVATIVE CONCEPTS

File-n-File 10	3.50
File-n-File 15	8.25

COMMODORE

C-128 NEW	CALL
1971 Drive	19.95
1973 Drive	19.95
1980 Monitor	CALL
1670 Modem	CALL
C-64 Computer	CALL
1541 Drive	159
1541C Printer	199
1700 Monitor	199
Simon & Basic	24.75
Assembler 84	34.75
Super Expander	22.75

MICROPROSE (C-64)

Kennedy Approach	21.75
Crusade in Europe	21.75
Decision in Desert	21.75
Solo Flight	30.75
Nato Commander	20.75
Helical Ace	18.75
F-15 Strike Eagle	20.75
Helical Ace	18.75

CARDCO

Digital Camera	189.95
Printer Buffer	59.95
Numeric Keypad	54.95
C-64 2 Slot Board (84)	54.95
C-64 2 Slot Board (84)	54.95
8-Megabyte Hard Disk	39.95
Winter Now-84	29.95
Spring Now-84	29.95
File Now-84	29.95
Ac Now-84	29.95
Cable Now-84	29.95

EPYX

(C-64)	28.75
Foot Lead	28.75
Breakdown	28.75
Greatest Baseball	24.75
Summer Games	20.75

NOVATION

IBM 3001200 MS-DOS ext	319
IBM 3001200 CNAME ext	279
IBM 3001200200 ext	329
IBM 3001200200 MS-DOS ext	329
IBM 3001200 ext	319
IBM CNAME ext	329
Microvision 3001200	319

RACAL-VADIC

2430PC	549
2430PC	759
2430V	559
1200PC	329
380V	225
380PC	195

BRODERBUND

The Print Shop	28.95
Graphics Library	18.95
Graphics Library II	18.95
Kartlandia	18.95
Capitan Goodnight	18.95
Bank of Writer	32.75
Colorrunner	30.75
View of the Sun	24.75
Spacemaker	19.75
Whistler's Star	24.75
Whistler's Brother	18.75

SUB LOGIC (C-64)

Fight Simulator II	32.75
Night Mission Pinball	20.75

SSI (C-64)

Colonial Conquest	34.75
Wings of War	34.75
Computer Ambush	34.75
Field of Fire	34.75
Fighter Command	34.75
Kampfgruppe	34.75
Mech Brigade	34.75
Market Garden	34.75
St. Gun Shootout	34.75
Computer Baseball	34.75
Computer Quarterback	34.75
Imperium Galactum	34.75
Phantoms	34.75
Carmen & Cullford	34.75
50 Mission Crush	34.75
Question	34.75

SPINNAKER (C-64ROM)

Cosmic Life ROM	19.75
Jetbook	19.75
Alphabet Zoo	19.75
Up for Grabs	19.75
Delta Drawing	19.75
Up for Grabs	19.75
Wordcomp	14.75
Facemaker	14.75
Friction Fever	14.75

3.5" DISKETTES

DENNISON	
3 1/2" 560D 5 pak	14.95
3 1/2" 560D 10 pak	26.95

MAXELL

3 1/2" 560D 10 pak	29.95
3 1/2" 560D 10 pak	35.95

3 M

3 1/2" 560D 10 pak	26.95
3 1/2" 560D 10 pak	35.95

MONITORS

SAKATA

SG 1600 13" Green	99
SG 1600 13" Amber	109
SG 1600 13" Green TTL	119
SG 1600 13" Amber TTL	129
SG 1300 13" Color Comp	259
SG 1300 13" Color	259
SG 1300 13" Color	259
STST-Tri Stand	29

TAXAN

SG 1600 13" Green	99
SG 1600 13" Amber	109
SG 1600 13" Green TTL	119
SG 1600 13" Amber TTL	129
SG 1300 13" Color Comp	259
SG 1300 13" Color	259
SG 1300 13" Color	259
STST-Tri Stand	29

PRINCETON GRAPHICS

4012 12" Amber	449
4012 12" RGB	449
4012 12" RGB	449

TEKNIKA

MU 10 Composite	179
MU 10 RGB	225

PANASONIC

DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247
DT3000 13" RGB Composite	247

ZENITH

ZVM 120A Amber	75
ZVM 120B Green	75
ZVM 120C Amber BM	75
ZVM 120D RGB	75
ZVM 120E RGB	75
ZVM 120F RGB	75
ZVM 120G RGB	75
ZVM 120H RGB	75
ZVM 120I RGB	75
ZVM 120J RGB	75

AMDEK

300 Green	119
300 Amber	129
310 Amber IBM	129
Color 300 Audio	229
Color 300 Composite	369
Color 400	369
Color 700	469
Color 710	569

NEC

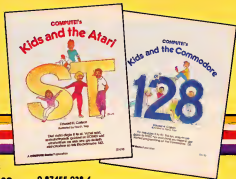
JP-1260 Green	99
JP-1260 Green	139
JP-1218 Color	239
JP-1218 Color	375
JP-1260 Color	375

LYCO COMPUTER

AMERICA'S MAIL ORDER HEADQUARTERS!

NEW

COMPUTE! Books For Kids



Help your children learn the basics of computer programming with these two new entertaining and educational books from COMPUTE!.

0-87455-038-6
\$14.95

0-87455-032-7
\$14.95

Each book contains easy-to-follow instructions, programming examples, quick reviews, and colorful illustrations. Written in COMPUTE!'s clear, easy-to-understand style, the books offer hours of entertainment while helping kids (and adults) learn to program in BASIC.

If you're acquainted with BASIC, you can easily write your own games and applications on Atari's ST or Commodore's 128 computers. Over 30 sections—all with instructor notes, lessons, assignments, and lively illustrations—entertain and amuse as you learn to use these powerful computers. COMPUTE!'s *Kids and the Atari ST* and COMPUTE!'s *Kids and the Commodore 128*, in the bestselling series from author Edward Carlson, are gentle introductions to programming your new computer. Clear writing and concise examples, both trademarks of this series, make it easy for anyone—child or adult—to learn BASIC painlessly.

Look for these and other books from COMPUTE!
at your local book store or computer store. Or order directly from COMPUTE!.

To order, call toll free in the US 1-800-346-6767 (in NY 212-265-8360) or mail the attached coupon with your payment to COMPUTE! Books, P.O. Box 5036, F.D.R. Station, New York, NY 10150.

Please send me the following COMPUTE! books. My payment is enclosed.

_____ **COMPUTE!'s Kids and the Commodore 128**, (032-7) \$14.95 each _____

_____ **COMPUTE!'s Kids and the Atari ST**, (038-6) \$14.95 each _____

ALL ORDERS
MUST BE
PREPAID IN
U.S. FUNDS

Subtotal _____
NC residents add 4.5% sales tax _____
Shipping and handling per book
(in U.S. and surface mail, \$5.00 per
book; airmail, \$6.00 per book.) _____
Total amount enclosed _____

- ☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Account No. _____ Exp. Date _____

Name _____ (Required)

Address _____

City _____

State _____ Zip _____

Please allow 4-6 weeks for delivery.

30412011

COMPUTE! Publications, Inc. 

One of the ABC Publishing Companies
525 7th Avenue 6th Floor New York, NY 10019

PUBLISHERS OF COMPUTE! COMPUTE! Gazette COMPUTE! Gazette DM COMPUTE! Books, and COMPUTE! Apple applications

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from Holt, Rinehart, & Winston, 55 Homer Avenue, Toronto, ON M8Z 4X6.

program could make the world different each time. Surprisingly, *Heart of Africa* offers no such option. There are very good reasons for this—the time limit, and the quest itself—but perhaps the game would be even more complete if each Africa could be a new one. Discoveries are less exciting when you know about them beforehand. Furthermore, the desperate feeling of being hopelessly lost, which *Seven Cities of Gold* presented so well, cannot happen here. If the game has a flaw, this is it.

But the flaw is easily overcome. The romance of uncovering the Dark Continent captures the imagination today as much as ever, perhaps because there remain no large, unexplored land masses anywhere in the world. *Heart of Africa* lets you canoe down the Congo, meet a Zulu chief, and even get caught in a whirlpool near Stanley Falls. Khartoum, Timbuktu, the Zambesi, Lake Tanganyika, Tangier—they're all there, waiting for you, ready to throw you many surprises.

An almost flawless development of an already excellent game system, *Heart of Africa* should excite anyone who found *Seven Cities of Gold* even remotely interesting. Now, if only I could find Dr. Livingstone.

Heart of Africa
Electronic Arts
2755 Campus Drive
San Mateo, CA 94403
\$32.95



On the trail of corporate skullduggery in Activision's *Hacker* (Commodore 64 version).

the computer indicates that a security malfunction has occurred. You're in.

The logon sequence is very realistic. Once, at the beginning of a game, someone walked into the room and watched me guess at a few passwords. Hearing that we were trying to get into an unknown system, which might be a government computer, and then seeing the security malfunction message, he got worried and reminded us that it's illegal to do what we were doing. That's the great appeal of *Hacker*, the feeling that you're doing something wrong and that you might get caught. Who knows, the FBI might even show up at your door and confiscate your computer.

After you enter the system, the game becomes less realistic. On an actual telecommunications system, everything would be straight text. The author of *Hacker*, in the interests of playability, has inserted some high-resolution graphics—unlike anything you'd see on a true bulletin board system or information service. However, the graphics do add a lot to the game.

Remote-Control Robots

You soon discover that you've come across a company involved in some sort of top-secret illegal project. This makes you feel less guilty about breaking into someone's system; you can seek out more information about this project and bring the culprits to justice.

The company owns a vast network of subterranean tunnels, and their computer (to which you've gained access) controls robots that travel through the tunnels. By using the robot to explore the tunnel network and occasionally coming to the surface, you can accumulate more details about the project. I'll say no more about the most effective techniques for winning because an important part of the game is figuring out what's going on.

The game play is almost identical in the versions I tried on the Commodore 64, Atari 520ST, and Amiga. The newer 16-bit machines (Amiga and ST) displayed slightly better graphics than

the 64 because their screens have higher resolution and more colors. The ST version works on both monochrome and color monitors.

All things considered, *Hacker* is a worthy addition to your software collection, especially if you enjoy adventure games that require a bit of thought and an investment of time.

Hacker
Activision, Inc.
2350 Bayshore Frontage Road
Mountain View, CA 94043
\$24.95 (Atari 400/800/XL/XE)
\$29.95 (Commodore 64/128)
\$39.95 (Apple and IBM)
\$44.95 (ST, Amiga, and Macintosh)

MasterType's Writer For Apple

Stephen Levy, Book Editor

Requirements: Apple IIc or Apple IIe with 128K RAM and a printer. A Commodore 64/128 version is scheduled for release this spring.

Does the world really need another word processor? After all, *MasterType's Writer* does all the things most word processors do. Using direct commands or on-screen menus, you can write, edit, save, search, move, change, find and replace, and print just as you can with most full-featured word processing programs.

So what makes *MasterType's Writer* special? If you're using it with an Apple IIc/IIe and an Imagewriter or Imagewriter II printer, and if you need multiple fonts—including some very large print styles—*Writer* is worth a closer look, even if you already have a word processor. With an Imagewriter or Imagewriter II, *Writer* can dump an exact copy of what's on the screen to the printer. *Writer* comes with eight fonts which can be loaded from disk and saved with your text. Among the styles are fonts that print very large type to the screen, quite suitable for use by young children just learning to read; proportionally spaced fonts of various sizes; and a style that is quite suitable for use on a monochrome monitor.

Each font can be edited, so you can modify those provided or design your own completely new font. And once created, you can use the screen dump feature to duplicate text written with the new font on paper.

If you have an Imagewriter II with a color ribbon, it's a simple matter to print text in color—simply underline

Hacker

Todd Heimarck, Assistant Editor

Requirements: Commodore 64 or 128; Apple II-series computer with at least 64K RAM; Atari 400/800, XL, or XE with at least 48K RAM; IBM PC/PCjr with at least 128K RAM; Atari ST; Amiga; or Apple Macintosh. Disk only.

When the first thing you see is the prompt LOGON PLEASE, you want to reach for the rule book. But apart from a card that tells you how to load and run this game, there are no instructions. None at all.

The premise behind *Hacker* is that you have stumbled across a telecommunications system about which you know nothing. Being a good hacker (if that's not an oxymoron), you feel the urge to break in and explore. Try a few passwords; unless you're very lucky, none of them will work. After several failures, the system logs you off and the game ends. Or does it? Some random characters appear on the screen, and

the text to be printed in green with a green line, blue text with the blue line, and so on. Again, you get an exact copy on paper.

Some Nice Touches

In addition to the fancy printing features, *MasterType's Writer* includes a few other extras. For example, the on-disk tutorial is well done and is a good introduction to using the program. Many people will return to the tutorial a second or third time even after they've started creating documents.

With *Writer's* dual windows, you can work on two documents at the same time. You can have an outline in one window and the text you're writing in the other. If you've never used this kind of feature before, you might not miss it; but once you've tried it, you'll wonder how you got along without it. *Writer's* dual windows have the added advantage of allowing you to decide how much of the screen each window will occupy at any time.

The manual is arranged in alphabetical order with entries for most of the terms you're likely to look up. Usually a term refers you to the appropriate instructions. If you're the type who likes to jump right in, you may find the manual a bit frustrating. But if you've tried the on-disk tutorial, you'll find the manual easy to use. And once you've been using *Writer* for awhile, an alphabetically arranged manual makes locating information a snap.

Another powerful feature of *MasterType's Writer* is keyboard macros—you can recall a series of instructions with one or two keystrokes. Macros are especially handy for storing a series of often-used words. If you're writing a book report, for example, you might need to type the author's name or the book's title many times throughout the report. By defining these phrases as macros, you can type them simply by pressing two keys.

Since macros can include program commands as well as ordinary characters, you can create macros for such purposes as saving your document on disk. Then, whenever you want to save the current copy of your work, you just press two keys.

Ease Of Use

MasterType's Writer gives you the choice of using direct commands—usually accessed by pressing CONTROL and one other key—or menus. Moving through the menus is easy and fast and saves you the trouble of memorizing commands. The menus are ideal for those new to word processing. Direct commands are faster for some functions, but for others save little more than one or two keystrokes. Most people will probably use a

combination of both menus and direct commands.

If you revise text often, one aspect of *MasterType's Writer* you may find annoying is its text entry and editing line. *Writer* doesn't allow full-screen editing; all text must be entered and edited on the bottom line of the current window. That means you must press the cursor keys to move the line you wish to edit to the bottom of the window. This isn't a problem when first entering text, but later, when editing, you can't see what comes immediately after the line you're trying to alter without continuously moving the text up and down.

For whom is *MasterType's Writer* most suitable? It should be strongly considered by those who have never used a word processor, teachers or students who plan to use it in schools, Apple users with an ImageWriter printer, or anyone who is unhappy with their current word processing program.

MasterType's Writer
Scarborough Systems, Inc.
55 South Broadway
Tarrytown, NY 10591
\$69.95

HabaWriter For The Atari ST

George Möller
Assistant Technical Editor

Requirements: Atari ST with at least 512K RAM and a compatible printer.

If you've been using *ST Writer*, the free word processor from Atari, but have been wishing for a program that supports the drop-down menus and windows of GEM, then *HabaWriter* is for you.



HabaWriter takes advantage of the GEM environment and includes all the features we've come to expect in a good word processor. Even more important, *HabaWriter* is easy to use. The instruction manual isn't very long—only 46 pages. If you're accustomed to other word processors, this may seem strange. Many programs have entire books devoted to their use, and sometimes it's necessary to enroll in special classes to become really proficient. Even though the size of the *HabaWriter* manual is small, all the information you'll need is there.

When you start up *HabaWriter*, you see a menu bar at the top of the screen with seven headings: Desk, File, Edit, Search, Format, Style, and Print. Just as on the GEM desktop, each menu instantly drops down when you point to it with the mouse. A click of the mouse button picks any selection on the current menu. Happily, *HabaWriter* is an intuitive program—the way you want to do something is probably the way *HabaWriter* requires you to do it. And the commands on the menus allow you to do just about everything you can think of.

If you prefer not to use the mouse, the ST's ten special function keys let you access most of *HabaWriter's* features. If you can't remember which key

THE EPSON LIBRARY FROM
MERIDYNE PUBLISHERS

NEW!

EPSON
CONNECTION:
ATARI

ALSO NEW

It's time to use your ATARI 800XL for more than playing games. Manage your household, succeed in school, do reports for work. This book shows you how. (L.E. Zeitz) \$16.95

EPSON Printers: Tips & Secrets (C. Hall & C. Smith) \$14.95

EPSON Guide to PC Communications (Baron) \$19.95

LOOK FOR THE EPSON LIBRARY AT YOUR FAVORITE BOOKSTORE OR ORDER FROM
YOU CAN ORDER FROM

Meridyne Publishers, Inc.
1000 10th Ave., New York, NY 10018
212-625-4141, TELEX: EPSONA

does what, just press the Help key. A window opens on the screen to display a convenient chart of the function key commands.

You can even use the Style menu to change the on-screen typeface from plain text to boldface or underline. Underlined text, however, is displayed with true underlining only on the monochrome monitor in high-resolution mode; it won't be underlined on the color monitor in medium resolution. But when you print your document, the text is underlined. (If you're using a printer that's capable of underlining, of course.)

Multiple Windows

HabaWriter lets you use the mouse for most functions that would require special commands with other word processors. For instance, you can define a block of text simply by dragging the mouse cursor over it. To delete the block, you would then select the Cut option from the Edit menu. Even though the text is erased off the screen, it's still kept temporarily in a clipboard and may be pasted back into the document wherever you wish.

HabaWriter uses screen windows for other purposes, too. You can open up to six windows to display more than one document at a time. Moving text between the windows is as easy as moving text within a document. And the size of the windows can be changed, just as with any GEM window. Files can even be combined, allowing you to work with longer documents by using the Paste Document option.

On-screen rulers let you easily set tab stops anywhere you want, and horizontal scrolling lets you create documents as wide as 132 columns. (Only 80 columns are displayed on the screen at a time.)

Using options on the Format menu, you can center text on the screen, align it to the right or left margins, or "justify" the text, just as professionally typeset pages appear.

HabaWriter's use of the GEM environment and its wide range of features make it one of the most attractive application programs to date for the Atari ST. You'll find it's a snap to give your correspondence and club newsletters a polished look—without much of the strain that's usually involved in learning how to use a new word processor.

HabaWriter
HabaSystems, Inc.
6711 Valleyview Avenue
Van Nuys, CA 91406
\$74.95

MUST LIQUIDATE! COMMODORE PLUS/4 COMPUTER SYSTEM AT BELOW DEALER COST!

NOTE: Monitor not available. The Plus/4™ can hook up to your TV.



Ideal for home or business! Perfect for programmers!

Factory new!

Factory warranted by Commodore®. Built-in software for word processing, data processing, spreadsheets and 128 color graphics!

Commodore® designed this Plus/4™ specifically for programmers and small businesses! And then they made it VERY EASY to learn and use for novices. Popular business software is available for a variety of purposes. For programmers, this machine has easy-to-use powerful commands and 60K of usable memory. Can hook up as many as four disk drives.

Team up the computer with our compatible units of famous brand, factory reconditioned and warranted **DISK DRIVE** and **DOT MATRIX PRINTER**! Sorry, we can't print the brand name of the disk drive and printer. But phone us Toll-Free and we can tell you. With Disk Drive for data storage and Printer for "hard copies", you'll have a complete system...at low liquidation prices. Order TODAY!

Units sold individually or in any combination you desire.

	Original List Price	YOUR COST
PLUS/4 COMPUTER Item H-1087-9035-001 Ship, hand. \$9.00	\$299.00	\$79
DISK DRIVE Item H-1087-3653-013 Ship, hand. \$8.00	\$269.00	\$149
PRINTER Item H-1087-3631-005 Ship, hand. \$7.00	\$200.00	\$119
TOTAL Ship, hand. \$23.00	\$768.00	\$347

COMPLETE SYSTEM WITH 20 FREE FLOPPY DISKS

Total Original List ... **\$877.00**

Liquidation
Priced At Only ... **\$339**

Item H-1087-9035-019 Ship, hand. \$19.00

Credit card customers can order by phone, 24 hours a day, 7 days a week.

Toll-Free: 1-800-328-0609

Your check is welcome! No delays in orders paid by check.

Sells outside continental U.S. are subject to special conditions. Please call or write to inquire.

C.O.M.B. Direct Marketing Corp.
Authorized Liquidator
14606 28th Avenue North
Mettus, WA 99021-1507

Additional Features of COMMODORE PLUS/4

Data base of 999 records. Computer holds 50 lines of text before it must be transferred to disk drive. Excellent terminal for use with modern 128 colors available graphics depth. Splice screen and windowing capabilities. Compatible with all Commodore hardware except parallel and daisy. NOT compatible with C64 software.

DISK DRIVE

Inelligent, high-speed External 5 1/4" floppy diskette recorder. 2K RAM, 768 ROM. Maximum storage of 170K formatted data. 35 tracks. Uses single sided, single density disk. Serial interface. Second serial port for chaining second drive or printer. Data transfer rate of 400 bytes per second.

DOT MATRIX PRINTER

Bi-directional 6 x 7 dot matrix impact printer. 60 characters per second. Rise upper and lower case letters, numerals and symbols. All EBT graphic characters. Standard friction feed. Maximum of 80 columns width, dot addressable. GEM, ASCII character codes. Original plus maximum of two copies. Paper weight, 6.5" to 8.5" Size: 13" W x 8" D x 3 1/4" H. Weight: 6 1/2 lbs.

C.O.M.B. Direct Marketing Corp. Item H-1087-14606 28th Ave. N., Mettus, WA 99021-1507

Send this following name (Minnesota residents add 6% sales tax. Allow 3-6 weeks for delivery. Sorry no C.O.D. orders.)

Send **COMPLETE SYSTEM** Commodore® Plus/4™ Disk Drive, Printer and floppy disk Item H-1087-9035-019 at \$339 each plus \$19 each for ship, handling.

Send **COMMODORE® PLUS/4™ COMPUTER** Item H-1087-9035-001 at \$79 each plus \$9 each for ship, handling.

Send **DISK DRIVE** Item H-1087-3653-013 at \$149 each plus \$9 each for ship, handling.

Send **PRINTER** Item H-1087-3631-005 at \$119 each plus \$7 each for ship, handling.

If my check or money order is enclosed (No delays in processing orders paid by check, thanks to TeleCheck.)

Charge ☐ MasterCard ☐ VISA ☐

Acc. No. _____ Exp. _____

PLEASE PRINT CLEARLY
Name _____

Address _____

City _____ State _____ ZIP _____

Phone _____ Sign Here _____

Loading And Linking Commodore Programs

Part 2

Jim Butterfield, Associate Editor

Are you running out of memory for your programs? You don't necessarily have to buy a bigger computer. This month's installment shows how a technique called chaining lets you break up a large program into smaller parts to work on a common task. The technique applies to all Commodore computers, with either disk or tape.

There are three major ways of connecting programs together. *Chaining* allows several programs to perform a job, each program continuing the work that a previous program has started. *Load linking* lets one program load another program, with the new program starting fresh on a new task. *Overlaying* lets a main program call in additional subroutines, data tables, or graphics information. This month we'll discuss chaining.

When one of a series of programs has completed its share of the work, it may chain to a following program to continue processing the data. In effect, several programs group together to create a bigger program. On Commodore computers, chaining works with disk or tape. It's more common with disk because the various programs can be brought in more quickly. If used with tape, you can arrange the programs sequentially on the cassette so little time is lost in searching for the next program. We'll use disk for the following examples, but they can be readily converted to tape.

Why Chaining?

The most obvious reason to chain programs is to save memory space. On small computers, there isn't enough room for big jobs. So the program is broken up into "chunks." Each chunk is small enough to fit into memory, each does a specific task, and together they do the whole job. Even on computers that seem to have lots of memory, you may need to resort to chaining to relieve congestion. For instance, even though the Commodore 64 begins with 38,911 bytes of free memory, arrays of data can quickly fill up much of this work area.

Sometimes program flow is an important reason for chaining. If a statistical program has been processing some data, it might ask the user to choose from several options (draw a graph, print the data, etc.). Depending on which option is chosen, it may be convenient to call in a selected program to do the next job.

In this way, the original program needn't be cluttered with code to cover all the possible options; instead, the options are handled by programs called in as required.

Likewise, it's possible to write a program that starts up in several different ways. In one case, it might collect the data it needs from DATA statements. Another time, it might require input from the keyboard. On still other occasions, it might compute the data, read it from a file, or detect it by external sensors. No matter. We'll start up whatever "acquisition" program is appropriate, and when the data is ready to be

processed, the computer can chain to a common processing program.

Chaining is also a worthwhile exercise which can force you to break your programs into well thought-out modules. Your program can't leap about at will, since it can only reach whatever is in the current module; and you must tie up loose ends before you go to the next unit. Each time you chain, FOR-NEXT loops are scrapped, subroutine RETURNS are canceled, and the DATA pointer is RESTORED. You must make sure that these program areas are tidy before you chain, since they will otherwise be lost.

Program Architecture

A major advantage of chaining is that you don't lose variables between programs. Values, strings, and arrays that have been worked out by a previous program are carried through to the next program segment. This is useful, but it also calls for careful handling—we don't want to mash these values inadvertently.

Figure 1 shows how programs, variables, and arrays lie in memory. The point marked start-of-BASIC is where the program starts in memory. Behind the program is a point called start-of-variables; beyond this point the computer stores variables and arrays.

You usually don't need to know the exact addresses of these memory points; the computer takes care of the housekeeping for you. String variables go into this area, too—although not the strings themselves, just three-byte descriptors

that say where the strings are located and how long they are. (More on this later.)

Suppose you have a large program that chains to a smaller program. Figure 2 shows this happening.

The variables don't move; behind the second program is wasted space that isn't used. This creates no problem when you run the program. However, after this kind of chaining has taken place, you should not **SAVE** the second program or you'll save the wasted area too (**SAVE** always stores from the start-of-BASIC point to just before start-of-variables).

Here comes the problem. Let's take the reverse situation: a small program that chains to a larger one. Figure 3 shows the difficulty that results.

The big program overwrites and destroys the variables created by the first, smaller program. To keep this from happening, our first program must be the biggest of the two, or at least the same size.

If several programs are chained together, this rule always applies. The first program must be as big or bigger than any other program. It sets the start-of-variables point, and it must set it high enough so that all following programs won't run into trouble (for more information on this point, see "Commodore Program Chaining," *COMPUTE*, December 1985). The Commodore 128 in 128 mode doesn't need to worry about this problem. Since it keeps variables in a separate memory bank, loading a new BASIC program can't harm them.

Strings And Descriptors

As noted earlier, the variable and array area holds string information (the descriptors), but not the strings themselves. There are two places where the actual strings might be, and it's important to know about them. Say that your program contains a line like this:

```
370 A$="GORILLA"
```

When this line executes, the computer makes an entry in the variable table showing that there is now a variable called **A\$**, that its length is seven characters, and that it is located at its present position in the program text itself. Except on the 128, the string is used from where it lies within the program. The computer decides that there's no point in making an extra copy of **GORILLA**; when it needs this string, it takes it from the BASIC program line. This type of string is called *static* because it never moves from its original location. Static strings can mean trouble if you chain programs: Since chaining replaces the original program text with a second program, all static strings—which exist only in the first program's text—are destroyed.

There's a second kind of string, and that's the one we must use here. If a program contains a statement like **INPUT A\$**, the string which is typed by the user must be stored somewhere. This is called a *dynamic* string; the computer stores it in a safe place where it won't be disturbed by chaining.

Dynamic strings are created in two ways: by **INPUT** or **GET** statements and by string manipulations (**LEFT\$**, **RIGHT\$**, **STR\$**, concatenation, and so on). It's simple to change a static string into a dynamic one. The statement **A\$="GORILLA" + ""** concatenates (adds together) the strings **"GORILLA"** and **""**. Since **""** is a null (empty) string, this statement really means "add nothing to the string **GORILLA**." Though the contents of the string don't change, the computer is convinced that we now have a new string which must be stored elsewhere in memory.

Again, the Commodore 128 in 128 mode doesn't need to worry about this problem. Strings are kept in a separate memory bank, and

Figure 1. BASIC program storage

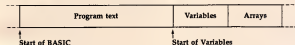


Figure 2. Chaining a smaller program from a large one.

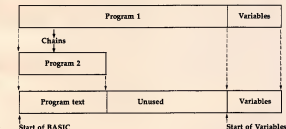
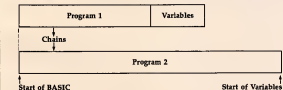


Figure 3. Chaining a larger program from a smaller one.



Chaining a longer program destroys the original program's variables.

there's no such thing as a static string in 128 mode.

Chaining Rules

Let's summarize the rules for well-chained programs:

- The first program in the chain must be as big or bigger than all subsequent programs.
- Any strings you need to pass from program to program must be dynamic, not static.
- If you use DEF FN definitions, redefine them in each program.
- Arrays should be DIMensioned only once, preferably in the first program.

A Short Example

Let's write a small series of programs to demonstrate how this works. Our first program is called MAIN:

```
100 IF N>0 GOTO 200
```

The variable N can only be zero when we start, so we won't jump ahead. But if we ever chain back to this program, we'll take the branch to line 200.

```
110 PRINT "SIMPLE GRADEBOOK DEMO"
120 DIM NS(15),M(15)
130 N=0
```

For simplicity, we'll assume eight students. When the program runs, you can invent their names and numeric grades.

```
140 FOR J=1 TO N
150 PRINT "STUDENT";J;
160 INPUT "NAME";NS(J)
170 INPUT "GRADE";M(J)
180 NEXT J
```

Running the program at this point gives you data on eight students. If you ever chain back to this original program, it will branch to line 200 (remember the IF test in line 100).

```
200 PRINT
210 PRINT "DO YOU WANT TO--"
220 PRINT "1. CALCULATE AVERAGE"
230 PRINT "2. CALCULATE HIGH/LOW SCORES"
240 PRINT "3. QUIT"
250 PRINT
260 INPUT "YOUR CHOICE (1-3)";C
270 ON C GOTO 300,310,320
280 GOTO 260
300 LOAD "C.AVG",0
310 LOAD "C.HIL",0
320 END
```

Note that line 300 will not run

into line 310, nor 310 into 320. The moment you perform LOAD within a program, the new program loads and runs immediately. Type this program and then save it as MAIN (don't save it under any other filename). Now type NEW and enter program C.AVG as follows:

```
100 PRINT
110 A=0
120 FOR J=1 TO N
130 A=A+M(J)
140 NEXT J
150 PRINT "AVERAGE SCORE,";N;
160 STUDENTS="";A/N
170 LOAD "MAIN",0
```

That's it. Check it closely and save it as C.AVG (again, the filename is important; don't change it). Now type NEW and enter program C.HIL as follows:

```
100 PRINT
110 H=M(1);L=M(1)
120 FOR J=1 TO N
130 IF H=M(J) THEN H=M(J)
140 IF L=M(J) THEN L=M(J)
150 NEXT J
160 PRINT "HIGH SCORE:";H;" BY"
170 FOR J=1 TO N
180 IF H=M(J) THEN PRINT NS(J)
190 NEXT J
200 PRINT "LOW SCORE:";L;" BY"
210 FOR J=1 TO N
220 IF L=M(J) THEN PRINT NS(J)
230 NEXT J
240 PRINT
250 LOAD "MAIN",0
```

Again, check your typing closely and save the program as C.HIL to complete the set. Now load program MAIN and you're ready to try out chaining. Note that MAIN is definitely larger than the other two. If there's any doubt in your mind, add some extra REM statements to MAIN to make it bigger.

Side Effects

We mentioned earlier that the act of chaining causes certain things to happen. FOR-NEXT loops are scrapped, subroutine RETURNS are canceled, and the DATA pointer is RESTORED. That makes sense: You can't RETURN to a program that has disappeared, for example. And occasionally, these side effects can be useful. For instance, can a program ever chain to itself? The answer is yes, but at first it's hard to see why you'd want to do so. What's the point of loading a program that's already there? The answer lies in these side effects.

Sometimes a program gets stuck deep in a subroutine and can't find its way out. With good programming, this should never happen. All subroutines should RETURN neatly, and if there's an error or similar anomaly, the information should be logged into a flag and detected at the appropriate program level. It's easy to give that sort of advice—but sometimes a program is deep within several nested levels of subroutines when the user commands, "Forget all this and take me back to the menu." Sensible programmers know that you can't jump directly out of these subroutines back to the main menu, and it's a long, long trail to back-track the whole way.

In case of emergency, you can chain the program to itself. As it loads itself back in, it shakes off all the FOR-NEXT loops and subroutine levels and surfaces cleanly—with all variables in place—at the first statement. Just to show it can be done, we'll write a dreadful program that does just that. Please don't write programs this way: It's here just to illustrate a point. Remember to type NEW before entering this program.

```
100 IF N>0 GOTO 130
110 PRINT "NAME LIST"
120 DIM NS(50)
130 PRINT
140 PRINT "DO YOU WANT TO --"
150 PRINT "1. ENTER NAMES"
160 PRINT "2. LIST NAMES"
170 PRINT "3. QUIT"
180 INPUT "YOUR CHOICE";C
190 ON C GOSUB 210,310,350
200 GOTO 130
210 PRINT "ENTER EACH NAME"
220 PRINT "FOLLOWED BY AN "CHARACTER"
230 PRINT "TO END ENTRY"
240 GOSUB 260
250 GOTO 240
260 INPUT NS
270 IF NS="" OR N=50 THEN LOAD "DEMO",0
280 N=N+1
290 NS(N)=NS
300 RETURN
310 FOR J=1 TO N
320 PRINT NS(J)
330 NEXT J
340 RETURN
350 END
```

Check the program and save it with the filename DEMO; be sure to use that filename, since the program uses it to load itself.

DEMO is a program turned bad, and you should try not to get yourself into a similar problem. By the time this program reaches line

210, it's in a subroutine; at line 260, it's nested within a second subroutine. When line 270 discovers that the user wants to exit, the poor programmer doesn't know how to get out. GOTO 130 would be a very bad solution: jumping out of the routine with GOTO instead of RETURN leaves unprocessed subroutine information on the computer's stack (which can eventually cause an OUT OF MEMORY error). What to do?

The second-best solution (shown here) is to clean up the program with a chain to itself. The best solution is not to get yourself into this kind of mess in the first place.

Chaining can be a useful and powerful technique. There are some rules to remember—especially that of making sure the first program is the biggest—but in general it works quite well. Don't confuse *chaining with loading*, where one program loads and starts another. In this case, there's no passing of variables; the new program starts clean. We'll talk about loading in next month's installment. ☐

Adding System Power To ST BASIC

Part 1

Kevin Mykytyn, Editorial Programmer

CONVERSE WITH YOUR COMPUTER

AT LAST! A FULL IMPLEMENTATION of the original ELIZA program is now available to run on your personal computer. Created at MIT in 1966, ELIZA has become the world's most celebrated and widely used intelligence demonstration program. ELIZA is a non-directive psychotherapist who evaluates each statement as you type it and then responds with her own comment or question—and her remarks are often amazingly appropriate!

Designed to run on a large mainframe, ELIZA has never before been available to personal computer users except as a greatly modified and somewhat lacking implementation which made the original program so fascinating.

Now, our new personal computer version possesses the FULL power and range of expression of the original, a being, offered at the introductory price of only \$49. And let us find out how she does it (or teach her to do more) we have included the complete SOURCE PROGRAM written in BASIC and no extra cost.

Order your copy of ELIZA today and you'll never again wonder how to respond when you hear someone say, "Okay, let's see what this computer of yours can really do!"

READ WHAT THE EXPERTS SAY ABOUT OUR VERSION OF ELIZA:
"Much more than a mere game... You'll be impressed with ELIZA. A convincing demonstration of Artificial Intelligence."
—PC MAGAZINE

"Delightful entertainment... An ideal medium for showing off your system."
—MICROCOMPUTING MAGAZINE

"ELIZA is an astounding piece of software... A fascinating program to use and study."
—SANDY'S MICROCOMPUTER REVIEWS

"ELIZA is a great way to introduce your friends to computers... A very fun party game."
—PETER A. MCWILLIAMS

"ELIZA is an exceptional program, one that's fun to use, shows off your machine, and has great financial value."
—POPULAR COMPUTING MAGAZINE

"This version of ELIZA is the best we have seen. As a party game, it's unmatched."
—HOME APPLICATIONS FOR THE C-64

ELIZA IS AVAILABLE IN THE FOLLOWING FORMATS:

- IBM PC, PCjr, PC-XT and all compatibles
- All Apple computers (II, II Plus, Ix, Ix2)
- Apple II (Macintosh BASIC required)
- Commodore 64 (specific disk or cassette)
- 5 1/4 inch or 5 1/8 inch disk for all CP/M systems

All versions are \$49 and exclude one page users manual.

Please add \$2.00 shipping and handling to all orders.

(California residents please add 6% sales tax)

ARTIFICIAL INTELLIGENCE RESEARCH GROUP
551 North La Jolla Avenue, Suite 100
La Jolla, CA 92036
(714) 966-7300 (714) 966-2724
VISA, M/C and checks accepted

Atari ST BASIC lacks commands for certain operations such as reading the mouse pointer, but it's possible to fill in these gaps by calling system routines with the VDISYS command. In Part 1 of this series, we'll examine the basics of calling VDI routines from BASIC and demonstrate a useful graphics routine. Part 2 will show how to read the mouse pointer with VDISYS and present a program for creating your own custom mouse pointers.

If you own an Atari ST, you've probably heard at least two of the three-letter acronyms associated with the computer. TOS stands for Tramiel Operating System—a huge system program which, at the most fundamental level, allows the computer to function. And GEM stands for Graphics Environment Manager, a separate system program that handles the ST's graphics-oriented desktop. GEM, in turn, consists of three separate parts: the VDI (Virtu-

al Device Interface), a low-level graphics interface that also handles mouse input; the AES (Application Environment Services), which uses the VDI to manage data and the desktop; and GEMDOS, which handles disk operations.

Interesting, you may say, but what's the point? For most BASIC programming, you needn't worry about TOS, GEM, VDI, AES, or GEMDOS, any more than the average driver needs to know exactly how an auto engine works. These system programs are the invisible machinery that makes everything else happen.

However, as you may have discovered, ST BASIC lacks commands to do certain tasks, such as drawing a circle or sensing the position of the mouse pointer. That's what makes one of these straggling programs—the VDI—an invaluable asset for the BASIC programmer. The VDI holds a treasure trove of system routines which can do everything from drawing boxes

and circles to rotating character fonts and manipulating raster blocks. With ST BASIC's VDISYS command, you can access all of these routines—which compensates in large part for the missing ST BASIC commands.

VDISYS To The Rescue

In simplest terms, the VDISYS command calls (activates) a VDI system routine to do a task that would be difficult or impossible to perform in BASIC. Furthermore, these system routines execute very quickly—a real plus when you're working with graphics. Whether executed in immediate or program mode, the VDISYS command always takes this general form:

VDISYS(x)

In this example a simple variable named *x* appears in the parentheses. It doesn't matter what value this variable represents; it's a *dummy* parameter, needed only to satisfy the syntax of the command. Don't try to enter this command yet—if you do, there's a good chance you'll see the mushroom cloud symbol that signals a system crash. A certain amount of preparation is always needed before you execute VDISYS.

When a VDISYS command is executed, control passes from your BASIC program to an internal VDI handler, which eventually passes control to the VDI routine itself. But first the VDI handler looks at certain sections of the computer's memory, called *parameter blocks*. The data in the parameter blocks tells the handler which particular VDI routine you want to execute. There's also other information that the VDI routine itself will need. If you don't supply all the information needed to call a routine, the VDI handler can't carry out your request.

VDI Opcodes

The first thing you must tell the computer is which VDI routine you want to call. Each VDI routine is identified by a unique *opcode number*. For instance, the VDI routine used in the program below has the opcode 11. This is a generalized shape-drawing routine. (There are hundreds of VDI opcodes and associated parameters, so we don't have

room in this article for a listing. But you can find a 42-page list of selected VDI opcodes in *COMPUTE!'s ST Programmer's Guide*, available from COMPUTE! Publications.)

Once you know a VDI routine's opcode number, that value must be **POKE**d into a special place in memory defined by the reserved variable **CONTRL**. Try typing **PRINT CONTRL** in immediate mode; even if you haven't given this variable any value, the computer prints a number on the screen. ST BASIC always predefines **CONTRL** along with several similar variables. The **CONTRL** variable represents an actual location in memory.

Since the system automatically substitutes this location for the keyword **CONTRL**, you don't have to memorize a series of numbers or worry about where this parameter block really resides. To select VDI routine 11, for instance, you simply execute **POKE CONTRL,11**.

How Many Corners?

Once you've **POKE**d the VDI opcode 11 into **CONTRL**, you must tell the computer how many vertices (corners) are needed to define the graphic shape you want to draw. Regular geometric shapes require different numbers of vertices. A triangle, for instance, requires a minimum of three corners. A rectangle, on the other hand, can be defined with only two—the upper left corner and the lower right one. Of course, a rectangle has a total of four corners, but the total is not what we're looking for. The computer cares only about the *minimum* number of vertices it takes to draw the shape in question. After you determine how many vertices are needed, that value is **POKE**d into the location defined by **CONTRL + 2**. For example, in line 30 of the program below, the statement **POKE CONTRL + 2,2** tells the computer that you want to draw a rectangle (defined by only two corners).

Note that the second **POKE** is directed two bytes higher in memory than the first. Now you can see the parameter block begin to take shape: It's simply a segment of memory where you place a collection of values. The first byte of the

parameter block is defined by **CONTRL**, and the remaining locations are defined as even-numbered offsets above that starting spot (**CONTRL + 2**, **CONTRL + 4**, and so forth).

The particular routine used in this program (termed a *generalized drawing primitive*) contains several subroutines (also called *subfunctions*), each of which performs a different drawing task. To choose a subroutine, you must **POKE** its identifying number (called the *primitive ID*) into the location defined by **CONTRL + 10**. In this case we want to use the bar-drawing subroutine, whose primitive ID happens to be 1. So in line 40 of the program, we **POKE CONTRL + 10,1**.

PTSIN And INTIN

The next step is to tell the VDI handler where to place the graphic shape. Recall that you told the computer earlier how many vertices it takes to define the shape. To position the shape on the screen, you must now tell VDI where to put each vertex. This is done by **POKE**ing horizontal (X) and vertical (Y) coordinate values into a second parameter block area.

The second parameter block begins at a memory location defined by the reserved variable **PTSIN** (*Points Input*). Again, you don't need to know the actual memory locations involved, since the computer keeps track of them for you. All you need to do is **POKE** the correct numbers into **PTSIN** (and even-numbered adjacent locations, in some cases).

Lines 50–80 of the example program perform this job by **POKE**ing the bar's X and Y coordinates into memory. The X coordinate of the first point is **POKE**d into **PTSIN**; the first point's Y coordinate goes into **PTSIN + 2**; the X coordinate of the second point goes into **PTSIN + 4**, and so on. Keep in mind that you must supply a *pair* of coordinate values for every point that you defined in **CONTRL + 2**.

A third parameter block, beginning at the address defined by the reserved variable **INTIN**, is used to pass *attribute values*, if any are required by the current subroutine. The term *attribute* is a catch-all that can include many different pa-

rameters—colors, rotation values, a style index, or whatever—depending on which subroutine is called. Since the subroutine used in this program requires no attributes, we don't need to POKE any values in this segment of memory. As a signal to the VDI handler that no attributes are involved, we must also POKE a zero into location CONTRL+6; this location tells the system how many attribute values to read from the INTIN parameter block.

After all of the required values have been POKED into memory, line 90 of the example program executes the VDISYS command, which calls the VDI routine and draws a bar on the screen. This may seem like an enormous amount of preparation for such a simple task (which some other computers can do with a single BASIC statement). On the other hand, it's better than not being able to draw a bar at all. You can cut down on the bulkiness of the code by writing setup subroutines that contain all the necessary overhead.

Bar Drawing

```
10 fullw 2:clearw 2:color 2,2,2
20 poke contrl,11 'VDI opcode
30 poke contrl+2,2 'number of vertices
35 poke contrl+6,0 'number of attributes
40 poke contrl+10,1 'primitive ID of bar
   command
50 poke ptsin,50 'x coordinate of top left
   corner
60 poke pstin+2,50 'y coordinate of top
   left corner
70 poke pstin+4,100 'x coordinate of
   bottom right corner
80 poke pstin+6,100 'y coordinate of
   bottom right corner
90 vdisys (0)
```

General Drawing Routine

Though every VDI call requires several preparatory steps, each individual step is easy to perform. As should be apparent by now, there's nothing mystical about the process—all you need to do is leave the right pieces of information in places where the computer can find them, then signal that you want the job done. The real work is done by the system itself.

Though the general procedure is the same in every case, each VDI routine requires different types and amounts of information. One of the

most useful VDI routines is the generalized drawing primitive used in the example program. Table 1 summarizes the POKES you need to call this routine.

Table 1: Generalized Drawing Primitive

```
POKE CONTRL, 11
POKE CONTRL+2, number of vertices
POKE CONTRL+6, number of attributes
POKE CONTRL+10, subfunction number
                    (primitive ID)
```

Again, CONTRL receives the opcode number of the VDI routine; CONTRL+2 the number of vertices in the desired shape; CONTRL+6 the number of attributes (if any); and CONTRL+10 the primitive ID for the subroutine you want. This particular VDI routine is extremely versatile and can draw pie-shaped segments, ellipses, filled or empty rounded rectangles, and other graphic images, including text. Table 2 lists the primitive IDs for each of this routine's subroutines.

Table 2: Drawing Subroutines

Primitive ID	Subroutine
1	bar
2	circle
3	arc
4	pie
5	ellipse
6	elliptical arc
7	elliptical pie
8	rounded rectangle
9	filled rounded rectangle
10	justified graphics text

To select a specific subroutine, find its primitive ID in the leftmost column of Table 2, then POKE that value into location CONTRL+10. Table 3 summarizes the POKES needed to set up the second and third parameter blocks (PTSIN and INTIN). Remember, the value POKED into CONTRL+2 (number of vertices) determines how many X-Y coordinate pairs you must POKE into the PTSIN parameter block. The X and Y coordinates for the first vertex go into PTSIN and PTSIN+2; the second X-Y coordinate pair goes into PTSIN+4 and PTSIN+6, and so forth.

Table 3: PTSIN And INTIN Parameter Blocks

```
POKE PTSIN, X coordinate of first vertex
                    (rectangle)
                    X coordinate of center
                    (circle, ellipse)
POKE PTSIN+2, Y coordinate of first
                    vertex (rectangle)
                    Y coordinate of center
                    (circle, ellipse)
POKE PTSIN+4, X coordinate of second
                    vertex (rectangle)
                    X radius for ellipse
POKE PTSIN+6, Y coordinate of second
                    vertex (rectangle)
POKE PTSIN+8, radius (circle only)
POKE PTSIN+12, radius (circular arc or
                    pie only)

POKE INTIN, start angle for arcs and pies
POKE INTIN+2, end angle for arcs and
                    pies
```

To draw a circle, ellipse, arc, or pie-shape segment, POKE X and Y coordinates for the shape's center point into PTSIN and PTSIN+2. A simple circle requires a radius value in PTSIN+8; arcs and pie shapes built from a part of a circle require a radius value in PTSIN+12. To draw an ellipse, or an arc or pie shape built from part of an ellipse, POKE the shape's X radius in PTSIN+4 and its Y radius into PTSIN+6.

Most of these subfunctions don't require any attribute values. To draw arcs or pie shapes, however, you must POKE two attribute values into INTIN and INTIN+2 to define starting and ending angles, respectively. Since the angle values are specified in tenths of a degree, not in whole degrees, these parameters can range from 0-3600. The starting angle specifies where you want the rounded portion of the arc or pie segment to begin, and the ending angle shows where that portion should stop. The statement POKE CONTRL+6,2 signals that you're passing two attribute values to the VDI.

As you'll learn from experimenting with these routines, VDISYS opens the gateway to a wide variety of graphics capabilities. Once you become familiar with the setup process, you'll probably find yourself using VDISYS more and more. In part 2 of this article, we'll look at VDISYS in more detail, and present a program that lets you create a custom shape for your ST's mouse pointer. ☐

Mousify Your Applesoft Programs

Part 2

Lee Swoboda

Part 1 of this series (COMPUTE!, March 1985) provided an Applesoft program allowing an AppleMouse, joystick, or game paddles to point to text on the screen. This month, Part 2 demonstrates more advanced mouse operations such as defining a text area and deleting, copying, or restoring the defined text. The example programs run with either DOS 3.3 or ProDOS. Although a mouse works best, you can substitute a joystick or game paddles.

Mouse-controlled programs must perform a number of functions in addition to simple pointing. The programs following this article provide several of these important capabilities:

- **Define Text.** Use the mouse to highlight a block of text, which can then be copied or deleted (typical word processing operations).
- **Copy Text.** Copy highlighted text to a buffer without deleting it from the screen.
- **Delete Text.** Delete highlighted text and save it in a buffer.
- **Insert Text.** Restore previously copied or deleted text at a new point on the screen.
- **Cancel.** Undo highlighting if you wish to abort a copy or delete operation.
- **Delete a Character.** Delete the character under the cursor.
- **Delete to End of Line.** Delete text from the cursor to the end of the line.
- **Find Mouse.** Locate the mouse interface.

Getting Started

Enter and save Program 1, which is an expanded and modified version

of the program published in Part 1. It works in either DOS 3.3 or ProDOS; if you're using ProDOS, change line 115 as shown here:

115 HIMEM: 36352

Program 2 creates a binary file named MOUSEY which contains machine language routines used by Program 1. The MOUSEY file created by Program 2 must be present on disk whenever you run Program 1. (It's not necessary to have Program 2 itself on the disk with Program 1, just a copy of the binary file created by Program 2.) Be sure to save a copy of Program 2 so you'll be able to create new copies of MOUSEY whenever needed.

Program 3 creates a short text file which we'll use in the following example. If you're using a joystick instead of a mouse, refer to the additional instructions under "Joystick Modifications" below. When you are ready to proceed, your disk should contain a copy of Program 1, a file named MOUSEY (created by Program 2), and a file named TEXT (created by Program 3).

When you run Program 1, the screen looks like this:

ENTER INFORMATION

FIRST NAME COMPUTE!
LAST NAME READER SERVICE
ADDRESS P.O. BOX 50950
CITY DES MOINES
STATE IA 50950
ZIP
TELEPHONE 1-800-346-6767

COPY DELETE INSERT CANCEL
ERASE QUIT DONE HELP

This screen simulates what you might see in a simple address book program. We have introduced an intentional error by putting the zip code entry on the same line as the

state entry. Let's correct the error for a quick demonstration of a few mouse features. Move the mouse cursor to the first number in the zip code, then press and hold the mouse button down while moving the mouse to the right. The computer highlights the zip code in inverse video. Keep moving the mouse until all the numbers in the zip code are highlighted, then release the mouse button.

At this point, the highlighted text area has been defined. Now move the mouse pointer to the word DELETE in the strip menu at the bottom of the screen and press the mouse button. The computer erases the highlighted zip code from the screen. Don't worry—the information hasn't been lost. Whenever you delete text, the program stores it in a temporary memory buffer.

Now let's put the zip code data back where it belongs. Move the mouse pointer to the beginning of the next screen line (directly under the I in IA), then press the mouse button. The computer moves the cursor to that line. Next, move the mouse pointer to the word INSERT and press the button again. The zip code data reappears in the desired screen area.

Mouse Editing Functions

Here is a more detailed description of the mouse-editing functions demonstrated in Program 1:

Mouse pointer and text cursor. The rapidly blinking caret symbol (') is the mouse pointer, which you can move around the text screen with the mouse. When the pointer passes over a character, the character blinks rapidly. The flashing rect-

angle shows the position of the text cursor. When the cursor passes over a character, the character changes temporarily to flashing uppercase. There are three different ways to move the text cursor:

- Move the mouse pointer to the spot where you want the text cursor to go, then press the mouse button.
- Use the arrow keys as you would in Applesoft BASIC (the Apple II uses CTRL-J and CTRL-K to move up and down, respectively).
- Press RETURN to move the cursor to the beginning of the next screen line. If the cursor is already on the bottom line, it moves to the top. Pressing RETURN does not erase the text to the right of the cursor.

Enter text. Text is entered as usual, by pressing any letter, number, or punctuation key. Lowercase letters are automatically converted to uppercase.

Define text. Before text can be copied or deleted, you must define it. Move the mouse pointer to the upper-left corner of the text you want to define, then press and hold the mouse button. While pressing the button down, drag the mouse pointer to the lower-right corner of the desired area. The computer marks the defined area by highlighting every character with inverse video. Now release the button: The area is defined, and you may proceed to the Cancel, Delete, or Copy options.

Delete text. To delete a text area that you previously defined, move the mouse pointer to DELETE in the strip menu at the bottom of the text screen, then press the button. The computer blanks out the highlighted portion of the screen and stores the first 200 characters of the defined area in a temporary buffer for later use.

Copy text. To copy a text area that you have previously defined, move the pointer to COPY in the strip menu, then press the button. The computer stores the first 200 characters of the defined area in a temporary buffer. Unlike the Delete operation, Copy does not blank out the defined area.

Insert text. To insert text that you previously copied or deleted, move

the pointer to the spot where you want to insert text, then press the button to locate the cursor at that spot. Now move the pointer to INSERT in the strip menu and press the button again. The computer inserts the text, using the text cursor position as a starting point. Note that the inserted text overwrites whatever else was in the affected area. You can insert only the most recently copied or deleted text.

Cancel. If you define a block of text and then decide not to copy or delete it, move the pointer to CANCEL in the strip menu and press the button. The highlighting disappears, and the text is no longer defined.

Editing keys. Press CTRL-D (or DELETE on the Apple IIc and IIe) to delete the character under the cursor. The remaining characters in that line move one space to the left. You can also press CTRL-X to delete every character from the present cursor position to the end of the line.

Try out the various editing functions. When you've tried everything, move the mouse pointer to DONE in the strip menu and press the button. The demonstration program ends with a routine that reads the current data directly from the video screen.

Since the Copy, Delete, Insert, and Cancel commands are written in BASIC, they may take a second or two to complete if you define a large text area. Though BASIC can't perform such operations very fast, these routines are far easier for you to customize than if they had been written in machine language. If the slowness bothers you, just imagine how long it would take to delete the same amount of text with your trusty pink eraser.

Joystick Modifications

If you don't own a mouse, you can substitute a joystick. Delete lines 120, 130, 10001-10090, 10200, 20220, and 44000-44050 from Program 1; then add or modify the lines in Program 4. The joystick moves the mouse pointer around the screen, and the joystick button substitutes for the mouse button.

Since the joystick was designed for a different purpose, its

performance doesn't equal that of a mouse. But it costs a lot less.

How The Program Works

The machine language routine contained in the MOUSEY file simply highlights text by changing every character between the text cursor and mouse pointer to inverse video. All the other functions are carried out by the BASIC routines in Program 1.

After you define a block of text, lines 35000-44050 act on the highlighted area. The Copy routine (36000-36180) converts each character in the defined area to normal video and stores it in a temporary text buffer. This buffer lies in locations 775-1000 (\$307-\$3E8), a normally unused region.

The Delete routine (37000-37180) is similar to Copy and uses the same temporary buffer, but replaces each character in the defined area with a blank space.

The Insert routine (38000-38100) moves text from the temporary buffer back to the video screen, beginning at the current location of the text cursor.

Lines 39000-40000 contain the Cancel routine, which aborts copy or delete operations. You can also cancel a definition by pressing any key.

The routine at lines 41000-41070 deletes a single character; lines 42000-42060 erase all or part of the current line.

Here are some other useful entry points in the program (note that each of these routines ends with a GOTO rather than GOSUB):

Line	Purpose
10120	read mouse
10300	position mouse pointer
10420	keyboard input
10570	position cursor

For instructions on entering these listings, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

Program 1: Advanced Mousification

```
#1 115 H1MEM: 37375
#F 120 EDSUB 44810
#1 130 #1 = 29: REM MOUSE SENSIT
      IVITY
#A 140 DS = CHRS (4)
#E 145 PRINT DS*"BLOAD MOUSEY"
#C 150 REM
#F 160 REM READ DATA FILE
#E 170 REM
#C 180 PRINT DS*"OPEN TEXT"
```

```

32 190 PRINT OS"READ TEXT"
40 200 INPUT NF$,NL$,AD$,C$,ST$
   ,Z1$,TE$
50 210 PRINT OS"CLOSE TEXT"
60 220 REM
70 230 REM DATA ENTRY SCREEN
80 240 REM
90 250 HOME
100 260 Y1 = 4: X1 = 15: C0 = 160
110 270 INVERSE
120 280 PRINT "          ENTER
   INFORMATION          "
130 285 VTAB 23: PRINT "    COP
   Y DELETE INSERT  CANCEL
   "
140 290 VTAB 24: PRINT "          E
   RASE QUIT DONE HELP
   "
150 300 NORMAL
160 310 VTAB 4: HTAB 1
170 320 PRINT "FIRST NAME ...
180 330 PRINT "LAST NAME ...
190 340 PRINT "STREET ...
200 350 PRINT "CITY ...
210 360 PRINT "STATE ...
220 370 PRINT "ZIP ...
230 380 PRINT "TELEPHONE ..."
240 390 VTAB 19: HTAB 10: INVERSE
   : PRINT "NORMAL
250 400 PRINT "IS MOUSE POINTER"
260 410 VTAB 21: HTAB 14: INVERSE
   : PRINT "NORMAL
270 420 PRINT "IS CURSOR"
280 430 VTAB 4
290 440 HTAB 15: PRINT NF$
300 450 HTAB 15: PRINT NL$
310 460 HTAB 15: PRINT AD$
320 470 HTAB 15: PRINT C$
330 480 HTAB 15: PRINT ST$
340 490 HTAB 15: PRINT Z1$
350 500 HTAB 15: PRINT TE$
360 510 REM *****
370 520 REM *****
380 530 REM *****
390 540 REM *****
400 550 REM *****
410 560 REM *****
420 570 REM *****
430 580 REM *****
440 590 REM *****
450 600 REM *****
460 610 REM *****
470 620 REM *****
480 630 REM *****
490 640 REM *****
500 650 REM *****
510 660 REM *****
520 670 REM *****
530 680 REM *****
540 690 REM *****
550 700 REM *****
560 710 REM *****
570 720 REM *****
580 730 REM *****
590 740 REM *****
600 750 REM *****
610 760 REM *****
620 770 REM *****
630 780 REM *****
640 790 REM *****
650 800 REM *****
660 810 REM *****
670 820 REM *****
680 830 REM *****
690 840 REM *****
700 850 REM *****
710 860 REM *****
720 870 REM *****
730 880 REM *****
740 890 REM *****
750 900 REM *****
760 910 REM *****
770 920 REM *****
780 930 REM *****
790 940 REM *****
800 950 REM *****
810 960 REM *****
820 970 REM *****
830 980 REM *****
840 990 REM *****
850 1000 REM *****
860 1010 REM *****
870 1020 REM *****
880 1030 REM *****
890 1040 REM *****
900 1050 REM *****
910 1060 REM *****
920 1070 REM *****
930 1080 REM *****
940 1090 REM *****
950 1100 REM *****
960 1110 REM *****
970 1120 REM *****
980 1130 REM *****
990 1140 REM *****
1000 1150 REM *****

```

```

10 30000 Y1 = 7: 80SUB 63050: C1 =
   A$
11 30070 Y1 = 8: 80SUB 63050: ST =
   A$
12 30080 Y1 = 9: 80SUB 63050: I1 =
   A$
13 30090 Y1 = 10: 80SUB 63050: TE =
   A$
14 30100 REM GO TO REMAINDER OF
   YOUR PROGRAM
15 30110 REM FOR EXAMPLE ...
16 30120 HOME
17 30130 VTAB 10
18 30140 PRINT NF$ "NL$
19 30150 PRINT AD$
20 30160 PRINT C1$, "ST$: "Z1$
21 30170 PRINT TE$
22 30180 CALL - 198: CALL - 198
23 30190 END : REM END OF EXAMPL
   E
24 35000 REM HIGHLIGHT TEXT
25 35010 POKE V0, C1 - 64: SW = 2
26 35020 IF SW > 1 THEN 10150
27 35030 IF X3 < X0 THEN X3 = X0
28 35040 IF Y3 < Y0 THEN Y3 = Y0
29 35050 POKE 770, Y3: POKE 771, X
   3
30 35060 POKE 772, Y0: POKE 773, X
   0
31 35070 CALL 37376
32 35080 Y3 = Y0: X3 = X0
33 35090 GOTO 10150
34 36000 REM COPY
35 36010 P3 = 775
36 36020 FOR I1 = Y2 TO Y3
37 36030 FOR J1 = X2 TO X3
38 36040 80SUB 40020
39 36050 C3 = PEEK (V2) + 128
40 36060 IF C3 < 160 THEN C3 = C
   3 + 64
41 36070 IF C3 > 223 THEN C3 = C
   3 - 64
42 36080 POKE V2, C3
43 36090 IF P3 > 1000 THEN 36120
44 36100 POKE P3, C3
45 36110 P3 = P3 + 1
46 36120 NEXT J1
47 36130 POKE P3, 141
48 36140 IF P3 < 1001 THEN P3 =
   P3 + 1
49 36150 NEXT I1
50 36160 POKE P3, 255
51 36170 SW = 0
52 36180 RETURN
53 37000 REM DELETE
54 37010 P3 = 775
55 37020 FOR I1 = Y2 TO Y3
56 37030 FOR J1 = X2 TO X3
57 37040 80SUB 40020
58 37050 C3 = PEEK (V2) + 128
59 37060 IF C3 < 160 THEN C3 = C
   3 + 64
60 37070 IF C3 > 223 THEN C3 = C
   3 - 64
61 37080 POKE V2, 160
62 37090 IF P3 > 1000 THEN 37120
63 37100 POKE P3, C3
64 37110 P3 = P3 + 1
65 37120 NEXT J1
66 37130 POKE P3, 141
67 37140 IF P3 < 1001 THEN P3 =
   P3 + 1
68 37150 NEXT I1
69 37160 POKE P3, 255
70 37170 SW = 0: C0 = 160
71 37180 RETURN
72 38000 REM INSERT
73 38010 P3 = 775
74 38020 I1 = Y1: J1 = X1
75 38030 C3 = PEEK (P3)
76 38040 IF C3 = 141 THEN I1 = I
   1 + 1: J1 = X1: P3 = P3 +
   1: GOTO 38030
77 38050 IF I1 > 22 OR J1 > 40 T
   HEN 38090
78 38060 IF C3 = 255 THEN SW = 0
   : C0 = PEEK (V0): GOTO 1
   0590
79 38070 80SUB 40020
80 38080 POKE V2, C3
81 38090 J1 = J1 + 1: P3 = P3 + 1
82 38100 GOTO 38030
83 39000 REM CANCEL
84 39010 FOR I1 = Y2 TO Y3
85 39020 FOR J1 = X2 TO X3
86 39030 80SUB 40020
87 39040 C3 = PEEK (V2) + 128
88 39050 IF C3 < 160 THEN C3 = C
   3 + 64
89 39060 IF C3 > 223 THEN C3 = C
   3 - 64
90 39070 POKE V2, C3
91 39080 NEXT J1, I1
92 39090 SW = 0
93 40000 RETURN
94 40010 REM
95 40020 V2 = 1023 + 128 * (I1 -
   1) + J1
96 40030 IF I1 > 8 THEN V2 = V2
   - 984
97 40040 IF I1 > 16 THEN V2 = V2
   - 984
98 40050 RETURN
99 41000 REM DELETE A CHARACTER
100 41010 80SUB 43010
101 41020 FOR I1 = V0 TO V2 - 1
102 41030 POKE I1, PEEK (I1 + 1)
103 41040 NEXT I1
104 41050 POKE V2, 160
105 41060 C0 = PEEK (V0)
106 41070 RETURN
107 42000 REM DELETE TO END OF L1
   NE
108 42010 80SUB 43010
109 42020 FOR I1 = V0 TO V2
110 42030 POKE I1, 160
111 42040 NEXT I1
112 42050 C0 = 160
113 42060 RETURN
114 43000 REM
115 43010 V2 = 1063 + 128 * (Y1
   1)
116 43020 IF Y1 > 8 THEN V2 = V2
   - 984
117 43030 IF Y1 > 16 THEN V2 = V2
   - 984
118 43040 RETURN
119 44000 REM FIND MOUSE
120 44010 FOR SW = 0 TO 6
121 44020 IF PEEK (49420 + (256 *
   SW)) = 32 AND PEEK (49
   659 + (256 * SW)) = 214
   THEN SW = SW + 1: RETU
   RN
122 44030 NEXT SW
123 44040 PRINT "I CAN'T FIND A M
   OUSE INTERFACE CARD" CH
   R$ (7) CHR$ (7)
124 44050 END
125 45000 REM #63000
126 45000 REM
127 45010 REM SUBROUTINE TO "READ
   "
128 45020 REM STRINGS FROM THE
129 45030 REM VIDEO BUFFER
130 45040 REM
131 45050 VTAB 24: FLASH : PRINT
   " WORKING ...
132 45060 " : NO
   RMAL : VTAB 1: HTAB 1
133 45060 A$ = ""
134 45070 REM CALCULATE V0
135 45080 REM (VIDEO BUFFER ADDR
   ES)
136 45090 V0 = 1037 + 128 * (Y1 -
   1)
137 63100 IF Y1 > 8 THEN V0 = V0
   - 984
138 63110 IF Y1 > 16 THEN V0 = V0
   - 984
139 63120 FOR I = 1 TO 25
140 63130 C0 = PEEK (V0 + 1)
141 63135 IF C0 < 128 THEN C0 = C
   0 + 128
142 63140 IF C0 < 160 THEN C0 = C
   0 + 64
143 63150 IF C0 > 223 THEN C0 = C
   0 - 64
144 63160 IF C0 = 160 AND PEEK (V
   0 + 1 + 1) = 160 THEN 6
   3190: REM END IF TWO BL
   ANKS
145 63160 IF C0 > 128 THEN C0 = C
   0 - 128
146 63170 A$ = A$ + CHR$ (C0)
147 63180 NEXT I
148 63190 IF RIGHT$ (A$, 1) = CHR$
   (32) THEN A$ = LEFT$ (
   A$, LEN (A$) - 1): GOTO
   63190: REM REMOVE TRAI
   LING BLANKS
149 63195 IF A$ = CHR$ (96) THEN
   A$ = ""
150 63200 RETURN

Program 2: MOUSE
Filemaker

16 10 REM BASIC PROGRAM FOR
17 20 REM GENERATING THE
18 30 REM BINARY FILE
19 40 REM "MOUSEY"
20 50 HOME
21 60 VTAB 12: PRINT "WORKING ..
   ."
22 70 FOR I = 0 TO 872
23 80 READ A$
19 90 POKE 37376 + I, A$
10 100 VTAB 12: HTAB 13: PRINT I
   + 1
11 110 NEXT I
12 120 PRINT CHR$ (4); "SAVE MOUS
   EY, A37376, 1873"
13 130 PRINT : PRINT "DONE!"
14 1000 DATA 173, 89, 170, 72, 165,
   217, 72
15 1001 DATA 165, 110, 72, 169, 2, 1
   33, 118
16 1002 DATA 169, 255, 133, 217, 16
   9, 191, 133
17 1003 DATA 51, 169, 0, 133, 243, 7
   6, 41
18 1004 DATA 146, 4, 0, 15, 0, 7, 8
19 1005 DATA 21, 0, 7, 0, 21, 0, 169
20 1006 DATA 29, 133, 133, 169, 146
   , 160, 0
21 1007 DATA 162, 11, 32, 77, 149, 7
   6, 69
22 1008 DATA 146, 28, 7, 32, 0, 8, 0
23 1009 DATA 22, 0, 8, 22, 0, 169
24 1010 DATA 57, 133, 133, 169, 146
   , 160, 0
25 1011 DATA 162, 11, 32, 77, 149, 1
   73, 0
26 1012 DATA 3, 141, 29, 146, 160, 0
   , 141
27 1013 DATA 30, 146, 173, 1, 3, 141
   , 31
28 1014 DATA 146, 169, 0, 141, 32, 1
   46, 173
29 1015 DATA 2, 3, 141, 33, 146, 169
   , 0
30 1016 DATA 141, 34, 146, 173, 3, 3
   , 141
31 1017 DATA 35, 146, 169, 0, 141, 3
   6, 146
32 1018 DATA 173, 4, 3, 141, 37, 146
   , 169

```

```

C 18198 DATA 8,141,38,146,173,5
  3
Z 18288 DATA 141,39,146,169,8,1
  41,48
B 18218 DATA 146,173,29,146,141
  61,146
C 18228 DATA 173,58,146,141,62,
  146,173
H 18238 DATA 62,146,285,34,146,
  48,15
I 18248 DATA 288,18,173,61,146,
  285,33
Z 18258 DATA 146,144,5,248,3,76
  134
H 18268 DATA 147,173,31,146,141
  63,146
Z 18278 DATA 173,32,146,141,64,
  146,173
H 18288 DATA 64,146,285,36,146,
  48,15
F 18298 DATA 288,18,173,63,146,
  285,35
H 18308 DATA 146,144,5,248,3,76
  123
F 18318 DATA 147,24,173,37,146,
  185,1
Z 18328 DATA 141,65,146,173,38,
  146,185
Z 18338 DATA 8,141,66,146,24,17
  3,39
E 18348 DATA 146,185,1,141,67,1
  46,173
B 18358 DATA 48,146,185,8,141,6
  8,146
B 18368 DATA 173,62,146,285,66,
  146,48
C 18378 DATA 18,288,29,173,61,1
  46,285
F 18388 DATA 65,146,176,21,173,
  64,146
E 18398 DATA 285,68,146,48,18,2
  88,11
Z 18408 DATA 173,63,146,285,67,
  146,176
H 18418 DATA 3,76,112,147,32,12
  2,148
E 18428 DATA 173,57,146,141,51,
  147,173
E 18438 DATA 58,146,141,52,147,
  173,144
H 18448 DATA 7,141,59,146,169,8
  141
Z 18458 DATA 68,146,173,68,146,
  281,8
Z 18468 DATA 48,9,288,24,173,59
  146
H 18478 DATA 281,128,176,17,24,
  173,59
F 18488 DATA 146,185,128,141,59
  146,173
E 18498 DATA 68,146,185,8,141,6
  8,146
H 18508 DATA 173,57,146,141,118
  147,173
H 18518 DATA 58,146,141,111,147
  173,59
Z 18528 DATA 146,141,144,7,238,
  63,146
F 18538 DATA 288,3,238,64,146,7
  6,195
E 18548 DATA 146,238,61,146,288
  3,238
E 18558 DATA 62,146,76,168,146,
  173,29
H 18568 DATA 146,141,61,146,173
  38,146
Z 18578 DATA 141,62,146,173,62,
  146,285
H 18588 DATA 38,146,48,15,288,1
  8,173
H 18598 DATA 61,146,285,37,146,
  144,5
C 18608 DATA 248,3,76,119,148,1

```

```

  73,31
F 18618 DATA 146,141,63,146,173
  32,146
H 18628 DATA 141,64,146,173,64,
  146,285
H 18638 DATA 48,146,48,15,288,1
  8,173
H 18648 DATA 63,146,285,39,146,
  144,5
H 18658 DATA 248,3,76,188,148,3
  2,122
F 18668 DATA 148,173,57,146,141
  228,147
C 18678 DATA 173,58,146,141,221
  147,173
C 18688 DATA 28,7,141,59,146,16
  9,8
H 18698 DATA 141,68,146,173,68,
  146,281
Z 18708 DATA 8,48,28,288,9,173,
  59
H 18718 DATA 146,281,127,144,19
  248,17
H 18728 DATA 56,173,59,146,233,
  64,141
F 18738 DATA 59,146,173,68,146,
  233,8
H 18748 DATA 141,68,146,173,68,
  146,281
H 18758 DATA 8,48,28,288,9,173,
  59
K 18768 DATA 146,281,64,144,19,
  248,17
H 18778 DATA 56,173,59,146,233,
  64,141
F 18788 DATA 59,146,173,68,146,
  233,8
H 18798 DATA 141,68,146,173,68,
  146,281
H 18808 DATA 8,48,28,288,9,173,
  59
E 18818 DATA 146,281,64,144,19,
  248,17
H 18828 DATA 56,173,59,146,233,
  64,141
H 18838 DATA 59,146,173,68,146,
  233,8
H 18848 DATA 141,68,146,173,57,
  146,141
H 18858 DATA 95,148,173,58,146,
  141,96
H 18868 DATA 148,173,59,146,141
  28,7
F 18878 DATA 238,63,146,288,3,2
  38,64
H 18888 DATA 146,76,181,147,238
  61,146
H 18898 DATA 288,3,238,62,146,7
  6,146
C 18908 DATA 147,76,11,149,56,1
  73,61
E 18918 DATA 146,233,1,141,57,1
  46,173
H 18928 DATA 62,146,233,8,141,5
  8,146
C 18938 DATA 169,8,133,138,169,
  128,174
E 18948 DATA 58,146,172,57,146,
  32,31
C 18958 DATA 149,142,58,146,146,
  57,146
Z 18968 DATA 24,169,255,189,57,
  146,141
F 18978 DATA 57,146,169,3,189,5
  8,146
F 18988 DATA 141,58,146,24,173,
  57,146
H 18998 DATA 189,63,146,141,57,
  146,173
F 19008 DATA 58,146,189,64,146,
  141,58
C 19018 DATA 146,173,62,146,281
  8,48

```

```

C 19028 DATA 28,288,9,173,61,14
  6,281
H 19038 DATA 8,144,19,248,17,56
  173
H 19048 DATA 57,146,233,216,141
  57,146
F 19058 DATA 173,58,146,233,3,1
  41,58
E 19068 DATA 146,173,62,146,281
  8,48
F 19078 DATA 28,288,9,173,61,14
  6,281
F 19088 DATA 16,144,19,248,17,5
  6,173
E 19098 DATA 57,146,233,216,141
  57,146
H 19108 DATA 173,58,146,233,3,1
  41,58
H 19118 DATA 146,96,184,133,118
  184,133
C 19128 DATA 217,184,141,89,178
  169,141
H 19138 DATA 141,1,2,169,1,133,
  52
H 19148 DATA 96,133,137,132,135
  134,136
H 19158 DATA 169,8,133,133,133,
  134,78
H 19168 DATA 136,182,135,144,13
  24,165
E 19178 DATA 137,181,133,133,13
  3,165,138
C 19188 DATA 181,134,133,134,6,
  137,58
H 19198 DATA 138,165,136,5,135,
  288,227
H 19208 DATA 164,133,166,134,96
  133,134
E 19218 DATA 132,135,168,8,169,
  8,145
H 19228 DATA 133,288,288,2,238,
  134,138
Z 19238 DATA 288,4,198,135,48,4
  282
H 19248 DATA 76,83,149,96,8,8

```

Program 3: TEXT Filemaker

```

10 DS = CHR$(4)
20 PRINT DS;"OPEN TEXT"
30 PRINT DS;"WRITE TEXT"
40 PRINT DS;"COMPUTE"
50 PRINT DS;"READER SERVICE"
60 PRINT DS;"P.D. BOX 58958"
70 PRINT DS;"DES MOINES"
80 PRINT DS;"IA 58958"
90 PRINT DS;" "
100 PRINT DS;"1-800-346-6767"
110 PRINT DS;"CLDSE"

```

Program 4: Keylock Modifications

```

10 265 DS = 4
20 18158 X8 = PDL (8)
30 18160 Y8 = PDL (1)
40 18161 B1 = PEEK (- 16287)
50 18162 IF B1 < 128 AND B8 = 3
  THEN B8 = 4
60 18163 IF B1 < 128 AND B8 = 2
  THEN B8 = 3
70 18164 IF B1 < 128 AND B8 = 1
  THEN B8 = 3
80 18165 IF B1 > 127 AND B8 = 2
  THEN B8 = 1
90 18166 IF B1 > 127 AND B8 = 4
  THEN B8 = 2
10 18170 IF PEEK (- 16384) > 12
  7 THEN 18448
11 18180 Y8 = INT (Y8 / 18) + 1
12 18198 X8 = INT (X8 / 18) + 1
13 28858 REM

```


IBM Variable Snapshot

Tony Roberts, Production Director

This programming utility lets you list the current values of all active variables in any BASIC program—an invaluable aid for debugging. It works on any IBM PC with BASIC or PCjr with Cartridge BASIC.

When things go haywire with a BASIC program, my first inclination is to check the variables: PRINT A\$, PRINT SCORE, PRINT UPPERLIMIT, and so on. Comparing what's actually stored in a variable with what you expected often helps to isolate programming problems.

Printing variable values over and over, however, quickly becomes tedious, especially when arrays are involved. "IBM Variable Snapshot" takes the work out of this process.

After temporarily appending the Variable Snapshot utility to your program, you can activate it with a simple GOTO command whenever your program stops with an error or you press the BREAK key. Once activated, Variable Snapshot sifts through memory, printing out first the scalar variables, then the array variables it finds there. Within seconds, you can see the values of all the variables your program has used. This kind of analysis has many benefits:

- By frequently checking the

variable list, you reduce the possibility of "forgotten" variables.

- You can quickly spot typographical errors in variable names. If the list contains both FILENAME\$ and FILENAM\$, you'll realize something is wrong.

- By checking variable types as well as names, you'll notice if the list contains both TOTAL% (an integer variable) and TOTAL! (a single-precision variable)—another common source of errors.

How To Take Snapshots

Type in Program 1 below and save it on disk in ASCII format. If you type it in with the "IBM Automatic Proofreader," published elsewhere in this issue, the program is saved in ASCII format automatically. Otherwise, use the command SAVE "SNAPSHOT.ASC",A.

Program 2 lets you test Variable Snapshot to verify that it's working properly before using it with your own programs. To run a test, type in Program 2 and save it on disk in ASCII format. Then append Snapshot to it with the command MERGE "SNAPSHOT.ASC". Now type RUN. The test program initializes several variables, then stops. When you type GOTO 1000 (the starting line number of Variable Snapshot), the name and value of each variable is printed on the screen. You can press CTRL-NUM LOCK on the PC or Fn-Q on the PCjr to pause the display, or stop it

at any time by pressing CTRL-BREAK on the PC or Fn-BREAK on the PCjr.

If the variable values are not what you expected, recheck your typing, paying close attention to the type declaration symbols (% , \$, ! , #) attached to the variables. If even one of these symbols is incorrect or missing, you'll have problems.

The test program initializes integer, string, single-precision, and double-precision variables as well as a full set of array variables. If everything prints out as expected, you can be pretty sure that Variable Snapshot is working well.

Friendly Filename And Quick Start

When Snapshot begins its work, the first thing it prints is the active disk filename, which the IBM stores in the 11 memory locations beginning at 4F1h (1265 decimal). This has nothing at all to do with variables, but simply provides an answer to the question "What did I call this program the last time I saved it?"

If you want to get started with Snapshot quickly, you can omit the entire array processing section (lines 1590-2220) and change line 1280 to read:

```
1280 IF QARRAYON% THEN END
```

This abbreviated version of Snapshot lists only simple variables, but you can go back later and add the lines to handle the array

Variable

Q%,QQ%,QQQ%
QTYPE%
QLENLEFT%
QDIMMS%
QARRAYON%
QSTRLEN%
QBASE%
QS
QCHARS
QFILES
QNAME%
QVAR%
QARRAY%
QFREE%
QASIZE%
QVALUE%
QSTRPTR%
QPTR%
Q%MSIZE()

Description

loop counters
variable type
number of characters left in variable name
number of dimensions in array
flag indicating if array boundary passed
length of string variable
status of OPTION BASE command
for single- and double-precision conversions
builds active filename
active filename
name of variable being processed
memory pointer to current variable
start of array space
start of free space
size of current array
temporary storage for integer values
points to location of actual string
points to start of next element in array
size of array of type Q%MSIZE()

variables. The REMs in the program listing are not referenced by other lines, so you can safely omit them when typing the program.

After you have Snapshot working, edit line 1000 to suit your preferences for screen color, width, and so on.

You may want to renumber Snapshot so its line numbers won't interfere with those of your own programs. (Low line numbers were used in the listing to make entering the program easier.) Load the program into memory and use the command RENUM xxxxx, where xxxxx is Snapshot's new starting line number. Then save the program back to disk, again using the ASCII option so Snapshot can be merged other programs.

The version I use begins at line 60000, and I've programmed a function key to execute the command GOTO 60000. Whenever a program halts, I simply press Fn-6 to see the value of every variable.

Array Bases

IBM BASIC includes the OPTION BASE statement for defining the lowest-numbered element in an array. If a program contains the statement OPTION BASE 0, or if no OPTION BASE statement is included, all arrays start with a 0 element. An OPTION BASE 1 statement means that arrays begin with element 1.

Variable Snapshot must know which OPTION BASE is in effect to

display array values properly. Memory location 45Ch (1116 decimal) provides this information. PEEKing that address yields either a 0 or 1, indicating which base is selected.

The adjacent memory location, 45Dh (1117 decimal), is related but a little more specific. If no OPTION BASE command has been issued, 45Dh contains a 0; if OPTION BASE 0 has been executed, 45Dh contains a 1; and if OPTION BASE 1 has been executed, the location contains a 2.

Try changing line 10 in Program 2 to read OPTION BASE 0 and observe the effect when running Variable Snapshot.

Although IBM BASIC allows arrays of up to 255 dimensions, few programs make use of more than one or two. For this reason, Variable Snapshot does not include provisions for arrays with more than two dimensions. Additional loops can be added to handle more complex arrays, if necessary.

A Few Cautions

To be truthful, Snapshot does not list every variable—it ignores those

that begin with the letter Q. The Snapshot routine itself, you'll notice, uses only variables beginning with the letter Q. That keeps Snapshot's own variables from being printed along with those of your program.

If you're inclined to tinker with this routine, you must be careful about introducing new variables. Lines 1020-1040 initialize every variable used by the routine, effectively reserving space for them in the variable table.

Lines 1120-1140 determine the boundaries of the variable table, reference points the program cannot do without. If a new variable is added to the program after the boundary measurements are taken, confusion results; the boundaries move and Snapshot loses its way.

Although Snapshot works with most programs, there can be complications. If you've written your program to make use of all available memory, there won't be room in the variable table for Snapshot's own variables. You'll need to leave Snapshot about 300 bytes of workspace.

How Snapshot Works

As mentioned above, Snapshot reads the boundaries of the scalar variable area, the array variable area, and the free space area, then works its way through the variable areas byte by byte deciphering the information stored there. Once it reaches free space, its work is finished.

The IBM stores scalar variables as shown below.

Following the last character of the variable name is the value of the variable.

• An integer variable is stored in two bytes in the standard low byte/high byte format. The high bit of the second byte indicates the sign of the integer. If it is set, the integer is a negative number.

Byte 1 = type (2 = integer, 3 = string, 4 = single precision, 8 = double precision)
Byte 2 = first character of variable name
Byte 3 = second character of variable name
Byte 4 = number of characters remaining in variable name
Byte 5
... = rest of variable name (high bit set)

- String variable pointers are stored in three bytes. The first is the number of bytes in the string, and the second and third point to the address (either in the string pool or in the BASIC program area) where the string is stored.

- Single-precision variable values are stored in four bytes. The values of these bytes can be concatenated into a string, then converted into a single-precision number using the CVS function.

- Double-precision variables occupy eight bytes, which can be concatenated and converted as above using the CVD function.

Array variables are stored similarly, but there's some additional information between the end of the variable name and the actual beginning of the variable values.

Following the variable name are two bytes that indicate the total size of the array. The next byte holds the number of dimensions. That is followed by two bytes describing the number of elements in the last dimension. Then two bytes describe the number of elements in the next to last dimension, and so on, until each dimension in the array has been defined.

Finally, the values of the array variables follow, and are stored in the same manner as values for scalar variables.

Using this information, the program listing, the description of Snapshot variables found in the accompanying table, and the actual program output, you should be able to develop a good understanding of how BASIC treats your variables.

For instructions on entering these listings, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

Program 1: IBM Variable Snapshot

```

01 1000 DEF SIZE=SCREEN 0,0:WIDTH 80:CO
      LDR 7,0
02 1010 REM initialize variables
03 1020 G%:=G%+0:G0%:=G%+0:G%LEFT%:=0:
      G%TYPE%:=G%+0:G%ANAY%:=G%+0:
      G%STLEN%:=G%+0:G%GWS%:=G%+0:G%PMS%:=G%+0:
      G%PMS%:=G%+0:G%PMS%:=G%+0:G%PMS%:=G%+0:
04 1030 G%PMS%:=G%+0:G%PMS%:=G%+0:G%PMS%:=G%+0:
      G%PMS%:=G%+0:G%PMS%:=G%+0:G%PMS%:=G%+0:
05 1040 G%PMS%:=G%+0:G%PMS%:=G%+0:G%PMS%:=G%+0:
      G%PMS%:=G%+0:G%PMS%:=G%+0:G%PMS%:=G%+0:
06 1050 REM Get active filename
07 1060 FOR G%:=0 TO 10
      G%CH%:=CHR$(G%+65)
08 1070 IF G%CH%=" " THEN G%CH%=" "
      G%CH%:=CHR$(G%+65)
09 1080 IF G%CH%=" " THEN G%CH%=" "
      G%CH%:=CHR$(G%+65)

```

[illegible]

```

# 1779 IF GOINDEX=1 THEN PRINT GNAME$;
      "X";#ID(GSTR(IGZ,2);)*," ",#
      #QVALUE ELSE PRINT GNAME$;"X";
      #ID(GSTR(IGZ,2);)*," ",#ID(GSTR
      (IGZ),2);)*," ",#QVALUE'
# 1780 OPTR=OPTR+2
# 1790 NEXT QX
# 1800 IF GOINDEX=2 THEN NEXT GOGX
# 1810 GOTO 2240
# 1820 REM scaling arrays
# 1830 PRINT
# 1840 IF GOINDEX=2 THEN FOR GOGX=GWB
EX TO GOINDEXZ(2)+GBASEX-#
# 1850 FOR GO=GOBASEX TO GOINDEXZ(1)+
      GBASEX-#
# 1860 GSCALE=PEEK(OPTR)
# 1870 GSTPTR=PEEK(OPTR+1)+PEEK(OPT
TR+2)*256
# 1880 IF GOINDEX=1 THEN PRINT GNAME$;
      "#ID(GSTR(IGZ,2);)*," ",#
      #CHRS(14) ELSE PRINT GNAME$;
      "#ID(GSTR(IGZ,2);)*," ",#ID(G
STR(IGZ),2);)*," ",#CHRS(14)
# 1890 FOR GOGX=# TO GSCALE-1
# 1900 PRINT CHR$(PEEK(GSTPTR+GOX));
      ;
# 1910 NEXT GOGX
# 1920 PRINT CHR$(34)
# 1930 OPTR=OPTR+3
# 1940 NEXT QX
# 1950 IF GOINDEX=2 THEN NEXT GOGX
# 1960 GOTO 2240
# 1970 REM single precision arrays
# 1980 IF GOINDEX=2 THEN FOR GOGX=GWB
EX TO GOINDEXZ(2)+GBASEX-#
# 1990 FOR GO=GOBASEX TO GOINDEXZ(1)+
      GBASEX-#
# 2000 Q=
# 2010 FOR GOX=# TO 3
# 2020 Q=Q+CHR$(PEEK(OPTR+GOX))
# 2030
# 2040 NEXT GOX
# 2050 IF GOINDEX=1 THEN PRINT GNAME$;
      "#ID(GSTR(IGZ,2);)*," ",#
      #CVD(Q) ELSE PRINT GNAME$;
      "#ID(GSTR(IGZ,2);)*," ",#ID(GSTR
      (IGZ),2);)*," ",#CVD(Q)
# 2060 OPTR=OPTR+4
# 2070 NEXT QX
# 2080 IF GOINDEX=2 THEN NEXT GOGX
# 2090 GOTO 2240
# 2100 REM double precision arrays
# 2110
# 2120 IF GOINDEX=2 THEN FOR GOGX=GWB
EX TO GOINDEXZ(2)+GBASEX-#
# 2130 FOR GO=GOBASEX TO GOINDEXZ(1)+
      GBASEX-#
# 2140 Q=
# 2150 FOR GOX=# TO 7
# 2160 Q=Q+CHR$(PEEK(OPTR+GOX))
# 2170 NEXT GOX
# 2180 IF GOINDEX=1 THEN PRINT GNAME$;
      "#ID(GSTR(IGZ,2);)*," ",#
      #CVD(Q) ELSE PRINT GNAME$;
      "#ID(GSTR(IGZ,2);)*," ",#ID(GSTR
      (IGZ),2);)*," ",#CVD(Q)
# 2190 OPTR=OPTR+8
# 2200 NEXT QX
# 2210 IF GOINDEX=2 THEN NEXT GOGX
# 2220 GOTO 2240
# 2230 REM loc address of next var iab
le
# 2240 GARRAYENZ(1)=GVAR+GBV
R-GARLEFTX-QTYPE*4+ELB
E
R=GVAR+GBV+GBASIZ+GBLEFTX+6
# 2250 IF GVAR=GBARRY THEN GVARRYD
N#=1
# 2260 GVAR=GBV+GBFREE THEN END
# 2270 GOTO 1170

```

Program 2: Snapshot Demo

[illegible]

ATARI TEXTDUMP

Ralph Johnson

Here's a short, simple utility that quickly dumps a GRAPHICS 0 screen to a printer. It works with all 400/800, XL, and XE computers.

I've always wanted the capability to dump a copy of a text screen to my printer. I also wanted this capability available from BASIC. My efforts to find such a program were fruitless. So, the only solution was to write one myself.

There were several requirements I established: 1) It should be fast—written in machine language. 2) It should sit in a relatively safe location in memory, surviving system resets. 3) It should always be ready to do its job, whether called in direct mode or from within a program. 4) It should be easy to use.

The result is "Atari Textdump." You can make your own copy of Textdump by typing in the program listing below. It creates an AUTORUN.SYS file on disk that automatically loads Textdump into memory page 6 (address 1536) when you boot the system from that disk. To call the routine, make

sure your printer is online and enter this statement:

A=USR(1536)

This works in both direct mode or within a program.

If you don't have a disk drive, or if you don't want Textdump to load as an AUTORUN.SYS file, delete lines 10–1000 in Program 1 and substitute this new line 10:

```
10 FOR A=1536 TO 1724:READ B:POKE
A,B:NEXT A
```

Again, you can call Textdump as described above in either direct or program mode. You can also convert this version of the program into a module for use in your own programs.

If you like, you can modify Textdump to print a smaller portion of the GRAPHICS 0 screen. Simply POKE the desired number of rows you want to dump into memory location 1613.

For instructions on entering the listing, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE!

Atari Textdump

```
110 CLOSE #1
120 OPEN #1,0,0,"0:AUTORUN
.SYS"
```

```
130 FOR A=1 TO 6:READ B:PR
INT #1;CHR$(B);:NEXT A
140 FOR A=1536 TO 1724:REA
D B:PRINT #1;CHR$(B);:
NEXT A
150 CLOSE #1
160 DATA 255,255,0,0,188
,6
170 DATA 164,162,80,169,
3,157,66,3,169,0,157
,74,3,169,144
180 DATA 157,68,3,169,6,
157,69,3,169,0
190 DATA 141,143,6,157,7
3,3,169,255,157,72,3
,32,66,228,165
200 DATA 88,133,203,165,
89,133,204,162,0,140
210 DATA 0,24,177,203,10
5,32,157,147,6,232,2
4,165,203,105
220 DATA 1,133,203,165,2
04,105,0,133,204,173
,143
230 DATA 6,201,24,240,52
,224,40,240,4,224,40
,208,217,169
240 DATA 155,157,147,6,2
38,143,6,162,80,169,9
250 DATA 157,66,3,169,14
7,157,68,3,169,6,157
,69,3,169,0
260 DATA 157,73,3,169,25
5,157,72,3,32,86
270 DATA 228,162,0,224,0
,240,173,162,80,169,
12,157,66,3,32
280 DATA 86,228,96,0,88,
50,155,0,0,0
290 DATA 0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0
300 DATA 0,0,0,0,0,0,0,0
310 DATA 0,0,0,0,0,0,0,0
,0,0,0,0,155,0
```

AmigaDOS Batch Files

Charles Brannon
Program Editor

AmigaDOS is more than a console-driven disk operating system. By executing a sequence of AmigaDOS commands stored in a file, AmigaDOS takes on some of the characteristics of a programming language. Whether you want to simplify repetitive disk commands or create personalized custom commands, batch files further extend the range and flexibility of AmigaDOS.

No matter how easy it is to use a program, the most popular programs are those that give users more power. And although a program may have scads of powerful commands, the most powerful programs are those which let users put the commands together in new ways—in effect, to write programs.

Instead of forcing you to always issue commands one at a time, a programmable application lets you create a script of commands to customize the behavior of the program. Whether we're talking about word processing macros, spreadsheet templates, relational database languages, or advanced machine language, programmability is the real key to software power. If you feel limited by a certain range of commands, you can combine the commands in new ways to create personalized features, just as we combine the vocabulary of English words to create a wealth of literature. Why just read when you can write?

Scripts, Sequences, And Batches

AmigaDOS is more than just a disk operating system—it's a programmable system that can process lists of its own commands as well as individual commands. In effect, AmigaDOS is a simple disk-oriented programming language.

A list of AmigaDOS commands can be stored in a disk file variously known as a script, a sequence, or a batch file. The term "batch file" is most commonly used by those who work with PC-DOS, MS-DOS, and CP/M, which are also programmable disk operating systems. To keep things straight, we'll use "batch files" synonymously with "scripts" or "sequences."

Even if you don't program in BASIC or any other language, you may be interested in learning about AmigaDOS batch files. The batch file "language" is simply made up of the same AmigaDOS commands you've probably been using all along (see "Introduction to AmigaDOS," Parts 1 and 2, in the February and March 1986 issues of COMPUTE!). There are also a few AmigaDOS commands designed especially for batch files.

Creating and running batch files is easy. Using a text editor, you just type in a list of AmigaDOS commands. Then you save the list on disk under a filename. To run the batch file, you type EXECUTE filename at an AmigaDOS prompt. AmigaDOS reads the batch file and executes the list of commands, just as if you had typed them one by one yourself.

We won't cover some of the

more advanced features of batch files, useful only to advanced C and machine language programmers. Instead, we'll concentrate on the everyday utility of batch file programming.

A Quick Example

In a moment, we'll show how to create batch files with ED, the AmigaDOS full-screen text editor, but first there's a simpler way to create a short batch file. Enter this line at an AmigaDOS prompt:

```
copy * to Hello
```

(Note that AmigaDOS commands can be entered in uppercase or lowercase.)

Although nothing seems to happen, AmigaDOS is waiting for you to enter some lines. We'll use the ECHO command to display a friendly message. ECHO displays any text that follows it within quotation marks, just like the PRINT statement in BASIC. One difference is that if you want to ECHO only a single word, the quotes aren't necessary.

At an AmigaDOS prompt, enter the following text, pressing RETURN after each line:

```
echo "Hello"  
echo "I am your friend, the Amiga"  
echo "personal computer."
```

After the last line, press CTRL-\. This key is the one to the left of the BACKSPACE key. CTRL-\. tells AmigaDOS that you're finished, and that it should finish writing and close the file. This key represents EOF, for End Of File.

To confirm that you've typed the file correctly, enter:

```
TYPE Hello
```

You should see the same lines you typed. Now you can start this simple program:

EXECUTE Hello

This should print on the screen:

Hello!

I am your friend, the Amiga personal computer.

Using ED

It would be nice to have the Amiga actually speak this greeting. Rather than type in a whole new file, we'll use ED, the screen editor, to make the simple changes we're interested in. Enter:

ED Hello

This runs ED and also loads the batch file named Hello. When you start ED, you can give it the name of any file to edit. If the filename doesn't exist, it will be created; otherwise the file is automatically displayed on the editor screen. (Incidentally, AmigaDOS has another text editor called EDIT, but it's not as easy to use as ED.)

We'll make the Amiga speak the ECHO messages aloud by taking advantage of the system's built-in speech synthesis via the AmigaDOS SAY command (added to AmigaDOS version 1.1). To learn more about SAY, just enter SAY by itself to enter an interactive mode with on-screen instructions.

After you start ED by typing ED Hello, the batch file we previously entered should be on the screen, with the cursor at the beginning of the first line. ED is a full-screen text editor, so you can move the cursor anywhere within the file (but not past the last line). To insert some text, just start typing. The DEL and BACKSPACE keys can be used to delete characters.

Move the cursor to the second ECHO line and press RETURN. This inserts a blank line. Cursor up to the blank line and enter:

SAY HELLO!

You don't need to press RETURN at the end of the line, since you already did this to open up a line for typing.

Now cursor to the end of the file and type:

SAY I am your friend, the Amiga personal computer.

(Notice that SAY is the only Amiga-

DOS command that doesn't require you to enclose text containing spaces with quotes.) This is how your screen should look:

echo "Hello!"

say Hello!

echo "I am your friend, the Amiga"

echo "personal computer."

say I am your friend, the Amiga personal computer.

With the cursor at the end of the file, press the ESC key. An asterisk (*) should appear. Press the X key, then RETURN. This exits ED and saves your changes back to disk.

Finally, type EXECUTE Hello to try out your talking batch file.

Although these techniques are sufficient for simple editing, ED has dozens of editing commands. For example, CTRL-B (press CTRL and B at the same time) blanks out and deletes the line the cursor is on. ESC-J-RETURN joins two lines together. Space doesn't permit a discussion of all these commands, but if you like to experiment, refer to the abbreviated ED reference chart accompanying this article.

Startup-Sequence

A special AmigaDOS batch file, called the *startup-sequence*, is executed automatically when you boot up an AmigaDOS or Workbench disk by inserting it at the Workbench prompt. Startup-sequence normally just displays a message, then launches the Workbench and ends the command line interface.

To edit this batch file, enter:

ed s/startup-sequence

This runs ED and calls up the file "startup-sequence" from the S subdirectory. This subdirectory, which can also be accessed as the \$: device, is a convenient place for batch files. Just as AmigaDOS by default searches for AmigaDOS commands in the C subdirectory, the EXECUTE command first looks for a batch file in the S subdirectory. If AmigaDOS can't find the batch file in this subdirectory, it looks for it in the current directory. So no matter what your current directory is, you can always use your batch file if you place it in the S directory on your startup disk.

When you first load startup-sequence into ED, you'll see something like this:

Common ED Commands

Immediate Commands (hold down CTRL and press key):

CTRL-A Insert line at cursor position

CTRL-B Delete current line

CTRL-D Scroll text downward

CTRL-E Move cursor to top or bottom of screen

CTRL-N Delete character at cursor

CTRL-O Delete word or series of spaces

CTRL-U Scroll text upward

CTRL-Y Delete to end of current line

Extended Commands (precede by pressing and releasing ESC):

B Move cursor to bottom of file

R/string1/string2/ Exchange string1 to string2

EQ/string1/string2/ Exchange, but query first

F/string/ Find string

J Join current line with next line

Q Quit without saving text

T Move cursor to top of file

X Exit, save text

echo "Workbench disk. Release L!"

echo "

echo "Use Preferences tool to set date"

echo "

LoadWB

endcl > nil:

Since this message appears every time you start up your disk, you may want to change the ECHO statements for a personalized message. Likewise, if you'd rather use AmigaDOS instead of the Workbench, delete the last two lines. The "> nil:" sequence makes AmigaDOS throw away the output of a command; here, the message "CLI task 1 ending."

Startup-sequence is a good place to put personalized commands. For example, if you like to keep your command directory in RAM for speed and convenience, you could insert these lines above the LoadWB line:

mkdir ramc

copy c: ramc all quiet

ed ramc

This copies all of the AmigaDOS commands from the C subdirectory on the floppy disk into a C subdirectory on the RAM disk. It also changes the current directory to the C subdirectory in RAM, so any AmigaDOS commands you type from then on will be loaded

from RAM: instead of from the floppy. In effect, this turns AmigaDOS into a memory-resident DOS, with all commands intrinsic instead of extrinsic. AmigaDOS responds much faster this way. However, this also uses up quite a bit of memory, so you may want to copy only the commands you use frequently.

Another useful startup action is to set the date and time. You can always do this with the Preferences tool or by opening a CLI and using the DATE command. However, it can be more convenient to enter the date when you first turn on your Amiga, allowing all files subsequently saved to be stamped with the current date and time. Just insert this line into startup-sequence:

```
date ?
```

The ? operator can be used in place of the parameter of a command. Instead of specifying the date, ? prompts the user to enter the date. It also displays the template for the date command (TIME,DATE, TO=VER/K:). If you like, use ECHO to display your own prompt, and > nil: to discard the template:

```
echo "Please enter the date and time."
echo "DD-MMM-YY HH:MM:SS"
date > nil: ?
```

From then on, whenever you boot up from this disk, you'll respond to the prompt by typing something like this:

```
27-Jan-86 15:12
```

which automatically sets the system clock.

Variable Parameters

You can also send special options to your batch file. Use these options on the command line along with the EXECUTE command. Just as with variables in BASIC, you can manipulate these parameters symbolically.

Let's say you'd like a batch file that gives you complete information on a file. It uses LIST to display the information about the file, and TYPE to display the file. You would use a command like EXECUTE SHOW RODEO to display the file RODEO. Use ED SHOW or COPY * TO SHOW to create this batch file:

```
.KEY name
LIST <name>
TYPE <name>
```

.KEY (don't forget the leading period) sets up a name for substitution text. Whatever you typed on the same line with EXECUTE is substituted wherever you use <name>. You must use the angular brackets, or LIST and TYPE would look literally for the file "name."

After creating this batch file, type this at an AmigaDOS prompt:

```
EXECUTE SHOW S/STARTUP-SEQUENCE
```

The result is the same as if you had typed LIST S/STARTUP-SEQUENCE followed by TYPE S/STARTUP-SEQUENCE.

Other AmigaDOS commands let you check to see if the user has entered a specific string and check to see if a file exists. To prevent an error message, we can check to see if the file exists before we use LIST and TYPE:

```
.KEY name
IF EXISTS <name>
LIST <name>
TYPE <name>
ELSE
ECHO "<name> does not exist"
ENDIF
```

Notice the use of IF, ELSE, and ENDIF. Looks like Amiga BASIC, doesn't it? In fact, the AmigaDOS IF-ELSE-END IF commands function very much like BASIC's. When the IF condition is true, AmigaDOS executes the following statements; otherwise the following statements are ignored. ELSE executes the statements following it only if the preceding IF was false. ENDIF cancels conditional processing and returns to executing all commands.

Any Parameters Missing?

Here's how to use the IF EQ option to test for the existence of a command-line parameter. If there is no parameter, <name> is null, so "<name>z" is simply "z". We use NOT to reverse the test. If the parameter "<name>z" is NOT equal to "z", then we must have a command line parameter. (We can't just test IF <name> NOT EQ "", since EQ wants two parameters, and the null string "" is not a parameter, but the lack of one.)

```
.KEY name
IF <name>z NOT EQ z
LIST <name>
TYPE <name>
ELSE
ECHO "You didn't give me anything to
```

```
SHOW."
ENDIF
```

Although you can't use leading spaces in the actual batch file, it's easier to follow the IF-ENDIF structures when you use indentation. Just don't type in the leading spaces. This version of the batch file SHOW checks both for the existence of the filename and for the presence of the filename parameter:

```
.KEY name
IF <name>z NOT EQ z
IF EXISTS <name>
LIST <name>
TYPE <name>
ELSE
ECHO "<name> does not exist"
ENDIF
ELSE
ECHO "You didn't give me anything to SHOW."
ENDIF
```

You can use more than one parameter in the .KEY statement, just as many commands, such as DATE, accept two inputs.

If the user doesn't enter anything for the parameter, you can assign a default value using either .DEF or \$. If you use .DEF, the default phrase is used throughout the batch file. In this example, SHOW displays itself if you don't give it a filename.

```
.KEY name
.DEF s/show
LIST <name>
TYPE <name>
```

You can use \$ to substitute a default value only for the current substitution. Several batch commands may use the value in different ways, so each command may have its own default value. In the following example, LIST displays the whole directory if <name> is null, but TYPE types the file "TEMP" if <name> is null:

```
.KEY name
LIST <name>
TYPE <name%temp>
```

Labels And Branching

You can jump forward to a label with the SKIP command. You'd typically use SKIP along with an IF condition if you want to skip over a block of statements that shouldn't be executed if the IF was true. You declare the label with LAB. SKIP can't skip backward, only forward to a LAB statement. You can usually use IF and ELSE to accomplish the same thing, though.

```

KEY name
IF exists <name>
TYPE <name>
SKIP ToMyLou
ENDIF
ECHO "<name> doesn't exist."
LAB ToMyLou
echo "Finished."

```

An EXECUTE command can execute another batch file, or even itself. This permits backward looping to some degree. Nested batch files can be quite handy. You can test and debug individual batch programs, then execute them together from a master execute script:

```

EXECUTE Greeting
EXECUTE GetDate
EXECUTE Assignments

```

The individual files could themselves contain other EXECUTE references.

ASSIGNING Shortcuts

If you're using EXECUTE a lot, you may grow weary of typing it. You can always rename EXECUTE to something short like x, but other batch programs may contain EXECUTE statements, requiring you to rename it back. Instead, you can use the ASSIGN command to assign any filename to a device name.

ASSIGN x: sysvc/EXECUTE

You can now use x: whenever you want to use the EXECUTE command. (The prefix sysvc/ makes sure that EXECUTE can be found no matter what directory you're in.)

The device name you create should not conflict with an existing one. To get a list of the current assignments, just type ASSIGN. You may want to ASSIGN d: c:\list for a convenient and quick shorthand for directories (c: is synonymous with the C directory). You can then just type d: to get a LIST.

ASSIGN can be so handy for this kind of thing that you'll probably want to include your own sequence of ASSIGN commands within startup-sequence. If you put your ASSIGN statement within startup-sequence, you'll get these assignments for every session. Just remember that ASSIGN can only be used to attach a device name to a particular filename. ASSIGN d: "c:\list quick" doesn't seem to work. Although LIST is a filename in the c directory, the "quick" parameter is not part of the filename. ☺

Apple Disk Duper

Jason Coleman

Here's a program that lets you duplicate Apple disks quickly and conveniently. Though it can copy disks formatted for either DOS 3.3 or ProDOS, it must be run with ProDOS. It also requires 128K RAM.

Everyone knows the value of backing up disks. But how many of us take the time to make archive copies of important disks on a regular basis? "Apple Disk Duper" simplifies the process by making it possible to copy an entire disk in only two passes. It works on one- or two-drive systems with at least 128K RAM.

After typing in the program and saving a copy, simply run it and follow the instructions on the screen. Apple Disk Duper prompts you every step of the way.

Although the program runs only under ProDOS, it can copy DOS 3.3 disks as well as ProDOS disks. It works with any Apple Disk II-compatible drive, but not with the new 3½-inch UniDisk.

Apple Disk Duper

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```

11 100 FOR X = 768 TO 785: READ
Y: POKE X,Y: NEXT
11 110 DATA 32,0,191,129,9,3,176
,249,96,3,96,0,32,0,0,0,0
11 120 TEXT : HOME
11 130 VTAB 12: HTAB 12: PRINT "
DISK DUPLICATOR"
11 140 VTAB 20: HTAB 9: PRINT "(
HIT ANY KEY TO BEGIN)":
POKE -16360,0: GET ST$

```

```

11 150 HOME
11 160 VTAB 12: INPUT "ENTER NUM
BER OF DRIVES: ";ND$:ND =
VAL (ND$)
11 170 IF ND < 1 AND ND < 2
THEN 390
11 180 HOME : VTAB 12: PRINT "PU
T SOURCE DISK IN DRIVE 1"
11 190 IF ND = 2 THEN VTAB 17: P
RINT "PUT DESTINATION DIS
K IN DRIVE2"
11 200 VTAB 20: POKE -16360,0:
PRINT "PRESS ANY KEY TO M
AKE COPY.": GET AK$
11 210 FB = 0:MX = 3
11 220 FOR N = 1 TO MX
11 230 POKE 771,120
11 240 POKE 780,32: POKE 770,96
11 250 FOR I = FB TO FB + 55
11 260 P2 = INT (I / 256):P1 = I
- 256 * P2
11 270 POKE 782,P2: POKE 781,P1
11 280 CALL 760: POKE 780,PEEK
(780) + 2: NEXT I
11 290 IF N < MX THEN PRINT CHR$
(4) "SAVE/RAH/COPY"N,AS
2000,LS6FFF":FB = FB + 56
11 300 NEXT N
11 310 IF ND = 1 THEN VTAB 12: P
RINT "PUT DESTINATION DIS
K IN DRIVE 1": GET AK$
11 320 FOR N = MX TO 1 STEP - 1
11 330 POKE 771,129: POKE 780,14
2
11 340 IF ND = 2 THEN POKE 770,2
24
11 350 IF N < MX THEN PRINT CHR$
(4) "BLAD/RAH/COPY"N:
11 360 FOR I = FB + 55 TO FB STE
P - 1:P2 = INT (I / 256):
P1 = I - 256 * P2
11 370 POKE 782,P2: POKE 781,P1
11 380 CALL 760: POKE 780,PEEK
(780) - 2: NEXT I
11 390 FB = FB - 56
11 400 NEXT N
11 410 IF MX = 2 THEN 440
11 420 MX = 2:FB = 140: IF ND =
1 THEN VTAB 12: PRINT "PU
T SOURCE DISK IN DRIVE 1
": GET AK$
11 430 GOTO 220
11 440 HOME : VTAB 12: HTAB 15:
INVERSE : PRINT "COPY COM
PLETE": NORMAL : END ☺

```


Smooth-Scrolling Billboards For IBM

Paul W. Carlson

Do you want to leave a message on your computer screen that's sure to be noticed? Or would you like to create an eye-catching display in a shop window that effectively communicates your message to the public? The programs presented here let you easily produce smooth-scrolling billboards on the 40- or 80-column screen of your IBM PC (with color/graphics adapter and BASICA) or PCjr (with Cartridge BASIC).

To be really effective, a billboard program must smoothly scroll its message across the screen. Programs that jerk the letters across the screen are very hard on the eyes. The speed necessary for smooth scrolling can be achieved only by avoiding the routines in the BIOS (Basic Input/Output System) and writing directly to video memory. However, this can cause a problem when text is used in graphics modes—writing directly to video memory disrupts the character generator. As a result, small flickering lines appear on the screen (for more details, see *COMPUTE! Books' Mapping the IBM PC and PCjr*, pages 193-198).

This problem can be solved by writing to video memory only during the time when the monitor's raster beam is in vertical retrace, while the display is idle. On some IBM-compatible computers (the

Compaq, for example), the problem can be avoided by writing to an inactive page of video memory and then making it the active page. The programs following this article make use of both methods.

With some computer and graphics card combinations, a few flickering lines remain at the very top of the screen when running the 80-column billboard program. These could have been eliminated, but only at the expense of speed and smoothness. About 300 characters can be written to video memory during the vertical retrace period, and 640 characters (eight lines of 80) need to be written for each screen update. Therefore, to eliminate the flickering lines entirely, we'd have to wait for three vertical retrace periods. These lines are less objectionable than the loss of smoothness caused by waiting for an extra retrace period.

Creating Billboards

Program 1 is for creating billboards on the 40-column screen, and Program 2 is for the 80-column screen. Both programs are extremely easy to use. After typing RUN, simply enter any text string at the prompt. If you want your message to contain a comma, enclose the entire text string in double quotes. When you press ENTER, the message enlarges and begins scrolling. It can be stopped at any time by pressing the Q key.

The programs can be customized to suit your taste. The character that forms the large letters can be changed from a solid block to another character by changing the DATA statement identified in the listing. For example, to change the solid block to a smiling face, change the DB to 02 in line 300. You can also modify the scrolling speed by changing the two bytes identified in the listing (the second byte has 256 times the effect of the first byte).

How It Works

The techniques used here can be applied to any program that must update a text screen very rapidly, so a brief description of the steps involved may be useful.

1. Set up a buffer in memory equal in size to the block of characters to be written to the screen (8 × 80 for the 80-column billboard).
2. For each input character, access the character PEL map in ROM at FFA6:0E. By columns, depending on whether or not a bit is set, put the code for a solid block or a space into the rightmost column of the buffer array.
3. When a column is complete, scroll the whole buffer one column to the left.
4. Wait for the beginning of a vertical retrace period, then copy the buffer to the inactive screen.
5. Make the inactive screen the active screen.

6. Do the next column in step 2.

For instructions on entering these listings, please refer to "COMPUTER's Guide to Typing in Programs" in this issue of COMPUTE.

Program 1: 40-Column Billboards

```

10 ' Forty Column Scrolling
   Billboard
20 '
30 ' Press the "Q" key to qu
   it.
40 '
50 DEF SEG: CLEAR, &HFFF0: N=&H4
   68A
60 FOR J=0 TO 249: READ A$
70 POKE N+J, VAL ("M"+A$): NEXT
   J
80 KEY OFF: CLS: SCREEN 0: WIDTH
   80
90 INPUT "Text string"; T$: T$=T
   $+ " "
100 N=&H4000: K=LEN(T$): FOR J=
   1 TO K
110 POKE N, ASC(MID$(T$, J, 1)):
   N=N+1
120 NEXT: POKE N, 0: CLS: WIDTH 4
   0
130 LOCATE, 0: N=&H468A: CALL N
140 WIDTH 80: CLS: KEY DN: END
150 DATA 00,00,EC,0C,00,0E,C0,
   09
160 DATA 00,02,0D,3E,00,41,1E,
   00
170 DATA 00,00,0E,00,0E,30,02,
   F3
180 DATA A4,1F,00,A6,FF,0E,C0,
   00
190 DATA 36,00,40,0A,1C,46,00,
   FB
200 DATA 00,74,F4,07,00,D1,E3,
   D1
210 DATA E3,D1,E3,03,C3,0E,09,
   00
220 DATA 00,33,FF,26,0A,07,00,
   05
230 DATA 00,41,47,43,E2,F5,56,
   06
240 DATA 09,09,00,51,33,FF,09,
   00
250 DATA 00,00,4E,00,00,05,00,
   41
260 DATA 72,04,00,20,00,02,00,
   0F
270 ' The following value is
   the
280 ' ASCII code of character
   that
290 ' forms the large text.
300 DATA 00
310 DATA 00,07,00,41,03,C3,50,
   47
320 DATA E2,EA,00,02,0E,05,BC,
   DB
330 DATA 0E,C3,FC,00,00,00,0D,
   36
340 DATA 00,41,0D,3E,00,41,09,
   4E
350 DATA 00,F3,A4,46,46,47,47,
   40
360 DATA 75,F4,A0,00,46,34,01,
   A2
370 DATA 00,46,04,05,50,A0,01,
   75
380 DATA 05,00,00,00,00,03,00,
   00
390 DATA 00,0E,C0,09,A0,00,0F,
   30
400 DATA 02,0D,36,00,41,0A,0A,
   03
410 DATA EC,AB,00,75,FB,EC,AB,
   00

```

```

420 DATA 74,FB,F3,A5,EB,04,EB,
   00
430 DATA EB,AA,09,00,00,0A,DA,
   03
440 DATA EC,AB,00,75,FB,EC,AB,
   00
450 DATA 74,FB,F3,A5,5B,CD,10,
   09
460 ' The following two value
   s are the
470 ' time delay constant in
   the order
480 ' least sig. byte, most s
   ig. byte.
490 DATA 01,00
500 DATA E2,FE,59,E2,DF,07,5E,
   04
510 DATA 06,02,FF,CD,21,3C,71,
   74
520 DATA 06,3C,51,74,02,EB,CF,
   00
530 DATA E5,07,00,00,05,CD,10,
   CB

```

Program 2: 80-Column Billboards

```

10 ' Eighty Column Scrolling
   Billboard
20 '
30 ' Press the "Q" key to qu
   it.
40 '
50 DEF SEG: CLEAR, &HFFF0: N=&H4
   68A
60 FOR J=0 TO 250: READ A$
70 POKE N+J, VAL ("M"+A$): NEXT
   J
80 KEY OFF: CLS: SCREEN 0: WIDTH
   80
90 INPUT "Text string"; T$: T$=T
   $+ " "
100 N=&H4000: K=LEN(T$): FOR J=
   1 TO K
110 POKE N, ASC(MID$(T$, J, 1)):
   N=N+1
120 NEXT: POKE N, 0: CLS
130 LOCATE, 0: N=&H468A: CALL N
140 WIDTH 80: CLS: KEY DN: END
150 DATA 00,00,EC,0C,00,0E,C0,
   09
160 DATA 00,05,0D,3E,00,41,1E,
   00
170 DATA 00,00,0E,00,0E,30,02,
   F3
180 DATA A4,1F,00,A6,FF,0E,C0,
   00
190 DATA 36,00,40,0A,1C,46,00,
   FB
200 DATA 00,74,F4,07,00,D1,E3,
   D1
210 DATA E3,D1,E3,03,C3,0E,09,
   00
220 DATA 00,33,FF,26,0A,07,00,
   05
230 DATA 00,41,47,43,E2,F5,56,
   06
240 DATA 09,09,00,51,33,FF,09,
   00
250 DATA 00,00,4E,00,00,05,00,
   41
260 DATA 72,04,00,20,00,02,00,
   0F
270 ' The following value is
   the
280 ' ASCII code of character
   that
290 ' forms the large text.
300 DATA 00
310 DATA 00,07,00,41,03,C3,A0,
   00
320 DATA 00,33,FF,26,0A,07,00,
   05
330 DATA 00,41,47,43,E2,F5,56,
   06
340 DATA 09,09,00,51,33,FF,09,
   00
350 DATA 00,00,7E,00,00,05,00,
   41
360 DATA 72,04,00,20,00,02,00,
   0F
370 ' The following value is
   the
380 ' ASCII code of character
   that
390 ' forms the large text.
400 DATA 00
410 DATA 00,07,00,41,03,C3,A0,
   00
420 DATA 47,E2,E9,00,02,EB,04,
   BC
430 DATA 00,0E,C3,FC,00,00,00,
   00

```

```

340 DATA 36,0A,41,0D,3E,00,41,
   09
350 DATA 9E,00,F3,A4,46,46,47,
   47
360 DATA 40,75,F4,A0,00,46,34,
   01
370 DATA A2,00,46,04,05,50,A0,
   01
380 DATA 75,05,00,00,00,EB,03,
   00
390 DATA 00,09,0E,C0,09,40,01,
   0F
400 DATA 00,04,0D,36,00,41,0A,
   DA
410 DATA 03,EC,AB,00,75,FB,EC,
   AB
420 DATA 00,74,FB,F3,A5,EB,04,
   EB
430 DATA BA,EB,AA,09,40,01,0A,
   DA
440 DATA 03,EC,AB,00,75,FB,EC,
   AB
450 DATA 00,74,FB,F3,A5,5B,CD,
   10
460 DATA 09
470 ' The following two value
   s are the
480 ' time delay constant in
   the order
490 ' least sig. byte, most s
   ig. byte.
500 DATA 01,00
510 DATA E2,FE,59,E2,DF,07,5E,
   04
520 DATA 06,02,FF,CD,21,3C,71,
   74
530 DATA 06,3C,51,74,02,EB,CF,
   00
540 DATA E5,07,00,00,05,CD,10,
   CB

```

COMPUTE!

**TOLL FREE
Subscription
Order Line**

1-800-247-5470

In IA 1-800-532-1272

STATE-OF-THE-ART MAGNETIC MEDIA

5 1/4" DISKETTES



- With Hub Rings
- Write Protect Tabs
- Envelopes
- User ID Labels
- In Factory Sealed Poly Packs of 10

(YOU GET EVERYTHING BUT THE BOX)

Prices are per Disk

QTY	50	100	500	1000
SSDD	.50	.50	.52	.49
DSDD	.64	.61	.57	.54

Library Case Holds 15 Diskettes \$1.00

plus \$0.04 S&H

The 100 Pk. Only \$10.99 plus \$2.00 S & H

100% ERROR FREE - LIFETIME WARRANTY

Min. order \$25.00 Add 10% for less than 50

diskettes. Shipping and Handling \$4.00 per 100

diskettes. Reduced shipping for larger quantities.

C.O.D. add \$4.00. Cash or certified check.

600

precision Data Products

P.O. Box 1101 Grand Rapids, MI 49501

(516) 452-1417 • Michigan 1-800-452-3448

Outside Michigan 1-800-258-0076

Commodore 64 Screen Genie

James A. Ledger

This thoughtfully designed utility helps you draw complete screens using character graphics. When you're finished, it writes a complete BASIC routine to recreate the graphics screen. The program runs on any Commodore 64 (or 128 in 64 mode) with either disk or tape.

"Commodore 64 Screen Genie" is both a screen editor and a program generator. With it, you can quickly and easily draw backgrounds for games, colorful title screens, or just pages of instructions. It offers a wealth of editing commands for designing a text or graphics character screen in normal, multicolor, or extended background mode. Then, almost instantly, it can write a BASIC routine to recreate that screen. This new routine is merged with whatever program is in memory. Since Screen Genie takes up no BASIC program space, it can be used with many other utilities such as the DOS Wedge, "TurboDisk," or "MetaBASIC."

Screen Genie is written entirely in machine language, so you'll need to enter it with the MLX machine language entry program found elsewhere in this issue. Follow the MLX instructions closely; here are the addresses you'll need for MLX:

Starting address: 0801
Ending address: 1D10

Built-In Help Screen

Screen Genie loads and runs like a normal BASIC program. Once you run it, however, the program

breaks into several modules which move to various places, leaving the BASIC program space completely free (more on this process later).

The first thing you'll see is a help screen showing all of the Screen Genie commands. Fortunately, you don't have to memorize all the commands shown here. Since the help screen is always available, the only key sequence you need to remember is CTRL-H (hold down CTRL and press H). Selecting any command from the help screen returns you to the work screen and performs that command. Pressing any other key simply returns you to the work screen. Of course, all of the commands are also available directly from the work screen.

The help screen serves another purpose by indicating which modes and cursor functions are selected. For instance, if you select the Paint cursor function by pressing the f3 function key, a white arrow appears next to that option on the help screen.

Once you enter the work screen, almost all of the keys work as they normally do—text and graphics characters can be typed in whatever color you like. However, you may not type a quotation mark, insert a character by pressing SHIFT-INST/DEL, or break out of the program by pressing RUN/STOP-RESTORE. The delete key (DEL) is not disabled, but works in a slightly different way: It erases the character at the cursor position and moves the cursor one space left, but it doesn't drag any characters on the right with it.

Finally, to prevent the screen from scrolling, you are not allowed to type anything in the bottom right corner. Instead, this space is used to show the current color for the characters you're typing. That's a handy feature, since the cursor itself is no longer a blinking box. Instead, it's a blinking black and white underline.

Immediate Commands

Screen Genie's commands are divided into four groups: immediate commands, cursor functions, screen modes, and color selection. Here is an explanation of the immediate commands:

CTRL-H (Help). Display help screen.

CTRL-T (Top clear). Clear from the top of the screen to the current cursor position.

CTRL-B (Bottom clear). Clear from the bottom of the screen to the current cursor position.

CTRL-M (Move). Move a block of characters from one screen location to another. Before you can move a block, you must first define its upper-left and lower-right corners. Press CTRL-M, then place the cursor on the upper-left corner of the block you want to move, and press RETURN. Move the cursor to the lower right corner of the block, then press RETURN a second time. Now the block is defined. To move it elsewhere on the screen, move the cursor to the place where you want to put the upper-left corner of the new block, then press RETURN. The contents of the new area are replaced by the contents of the defined block (note that the original

area is not disturbed). The Move command does not permit you to place the new block in any position that would overlap a screen border; all of the new block must fit inside the screen.

CTRL-Z (Memorize). Memorize the current screen by saving its contents in a memory buffer. A saved screen can be restored with CTRL-O.

CTRL-O (Oops). Swap the current screen with whatever is stored in the buffer. Pressing it again swaps it back. Besides restoring the screen after a manual save (CTRL-Z), this command can also undo any screen clear or move command.

CTRL-P (Program). Write a series of BASIC program lines to recreate the screen you've designed. These lines, beginning with the line number you choose, are merged with whatever BASIC program is in memory, if any. This feature performs a true merge, rather than simply tacking program lines onto the end of the current program. However, it does not replace any existing lines. If the merge operation would replace an existing program line, Screen Genie displays a message and gives you a chance to choose a new beginning line number.

The Program option also lets you add a line to set specific background and border colors. Likewise, if you're in extended background or multicolor mode when you choose this feature, you're given the option of adding lines that perform the setup for the current mode.

Finally, you have the option of adding a program line that waits for the user to press any key. This is useful for multiple pages of instructions, and so forth. Just be sure to include a prompt such as PRESS ANY KEY TO CONTINUE somewhere on the screen. If you choose this option while in extended background or multicolor mode, you may also add a line to turn the mode off after a key is pressed.

The default setting for all Program options is yes. Pressing any key other than Y or RETURN at the prompt selects no. If you have two or more sequential screens that use the same colors or mode, then you need only set these up on the first screen and turn the respective

mode off on the last screen.

CTRL-X (Exit to BASIC). This lets you save, load, and edit BASIC programs as usual. Screen Genie is designed so that you can exit to BASIC and later reactivate the utility without disturbing a BASIC program in memory. To reactivate Screen Genie, just type GENIE and press RETURN.

Under ordinary circumstances, pressing RUN/STOP-RESTORE does not disable Screen Genie. If you disable it in some other way, type SYS 50800 and press RETURN to start it up again. When you reenter Screen Genie, the work screen contains whatever was on the screen when you left BASIC: Press CTRL-O immediately to recall what you were working on at the time of your last exit.

Since the GENIE command works in program mode as well as direct mode, you can edit a previously designed screen by inserting the word GENIE just after the last PRINT statement, and then running only that portion of the program that displays the screen. For example, if the routine that recreates your screen uses lines 500-525, then you could add GENIE to the end of line 525 (or the beginning of line 526) and type RUN 500. You'll need to give the new screen a different beginning line number, and then delete the old routine when you exit. (Don't forget to remove the GENIE command from the program when it's finished.)

Modes

In addition to ordinary text mode (what you see when you turn on the 64), Screen Genie lets you work in extended background color mode or multicolor mode, or replace the usual character set with a custom-defined character set of your own. Consult the *Commodore 64 User's Guide* for additional information on how to use these modes.

CTRL-K (Extended background). This mode permits each character to have any of four different background colors, but lets you use only the first 64 characters of the character set.

CTRL-C (Multicolor). Since the ordinary character set looks quite strange in multicolor mode, this

mode will most likely require a custom character set. It cannot be used at the same time as extended background mode; selecting one mode turns the other off.

CTRL-U (User-defined characters). Selecting this mode causes the 64 to use a custom character set. Only the uppercase/graphics character set is available in this mode. Before choosing this option, you must store the character definitions in memory beginning at location 61440. Note that this configuration is only needed while you're editing the screen with Screen Genie. Once the screen design is done, and you have generated a BASIC routine to recreate the screen (see the Program option above), you can change your program to use whatever character set and memory locations you want.

Custom character mode demands a little more effort on your part. As in other cases, Screen Genie's Program option generates a complete routine with all of the necessary PEEKs, POKEs, and PRINTs needed to reproduce the screen. However, it's your job to put the custom character definitions in memory, decide on a location for the screen, and perform the extra POKEs needed to set everything up.

Cursor Functions

This group of options gives you additional control over the drawing cursor. They are selected by pressing one of the odd-numbered function keys. Any or all of these may be turned on at one time; however, if the Draw function (f1) is active, it takes precedence over the other three.

f1 (Draw with the cursor). This option lets you draw with any character. There are two ways to select the drawing character. You can either move the cursor to the desired character and press f1, or press f1 and type the character you want to use. To erase, press the space bar.

f3 (Paint with the cursor). Select a painting color just as you would normally change the cursor color in BASIC. Press CTRL or the Commodore key along with a number key from 1-8.

f5 (Change case with the cursor). This is very useful in extended

background mode where a shifted character has a different background color.

f7 (Reverse characters with cursor). This option is also handy in extended background mode, where reversing a character gives it a different background color. When you reverse a space character in normal mode, it has whatever color happens to be stored in color memory—unless the color happens to be the same as the background color, in which case it is changed to the current text color in order to make it visible. You can guarantee the color of reversed spaces by turning on the Paint function at the same time.

Color Control

The even-numbered function keys provide you with complete color control as follows:

f2. Cycle the border color (memory location 53280).

f4. Cycle the normal background color (location 53281).

f6. The menu lets you cycle background color registers one, two, and three (these color registers are used only in extended background or multicolor mode).

f8. Cycle the color of every character that is the same color as the character under the cursor. If you continue to press f8, Screen Genie remembers which characters you started changing and cycles only those characters, rather than switching to new ones each time. As soon as you press any other key, however, these characters are forgotten.

Compatibility

Screen Genie is designed to coexist with other Commodore 64 utilities as peacefully as possible. To minimize memory conflicts, nearly all of its program code and workspace areas reside in the hidden RAM under the 64's BASIC ROM, Kernal ROM, and I/O address space. Even so, some not-so-hidden RAM had to be used. The memory locations from 50800–52223 (\$C670–\$CBFF) are used for links to the system, interrupt-driven routines, sprite shapes, and screen memory. This still leaves locations 49152–50799 (\$C000–\$C66F) free for programs such as "TurboDisk," and locations

52224–53247 (\$CC00–\$CFFF) free for programs such as the DOS Wedge. Programs which reside in the upper BASIC program area, such as "MetaBASIC," will not be affected at all.

If you want to use other utilities with this program, install them *before* you load Screen Genie. There is one minor quirk when using Screen Genie with MetaBASIC. Screen Genie wedges itself in through the BASIC error vector at 768–769, which is reset by some of the commands in MetaBASIC. No harm is done when this occurs—you'll just have to reenter Screen Genie with SYS 50800 rather than the more convenient GENIE command.



The "Screen Genie" screen editor/program generator makes it easy to create graphic screens like this and incorporate them in your own Commodore 64 BASIC programs.

Screen Genie For Commodore 64

Please refer to the "MUX" article in this issue before entering the following listing.

```

0001:1B 08 8A 08 9E 32 30 37 B5
0009:34 3A 53 43 52 45 4E 70
0011:20 47 45 4E 49 45 00 0F
0019:00 A9 84 B5 0E A9 8D 85 23
0021:0F A9 80 85 10 A9 80 85 69
0029:11 A2 07 A8 00 88 B1 0E 29
0031:91 10 C0 80 D8 F7 E6 8F 69
0039:06 11 CA 30 06 D8 E8 A8 4F
0041:BB D0 EA A8 30 B9 8B 1C 4F
0049:99 67 CE B8 D8 F7 A8 00 00
0051:98 99 80 C7 00 D8 F7 A9 87
0059:FF B0 15 C7 A8 37 99 48 61
0061:C7 08 1A A8 15 A9 C8 96
0069:99 00 C7 88 88 88 18 F8 67
0071:A9 FF 80 80 C7 B0 83 C7 53
0079:80 D2 C7 8D 05 C7 A8 0F F4
0081:A9 83 99 C8 C7 B8 8B B8 6B
0089:18 F8 78 A9 22 85 81 A9 5C
0091:00 A8 80 A2 18 28 EF A8 84
0099:A9 26 B5 81 58 28 7B A8 84
00A1:18 20 92 A4 A9 81 D8 29 D9
00A9:18 A9 00 A4 A9 81 D8 29 D9
00B9:28 9B A5 3B 28 A9 A4 68 F5
00C1:85 40 20 C9 A9 12 A2 14
00C9:8A 20 9B A5 A9 CB A8 B8 1F
00D1:A2 84 20 EF A9 A9 D8 B5

```

```

00D9:BC A2 84 28 EF A9 A9 E1 D2
00E1:A2 09 28 9B A5 A9 80 8D A4
00E9:01 00 8D 02 08 A9 83 05 C8
00F1:20 85 2F B5 31 A9 0B B5 0D
00F9:2E 85 30 05 32 AD 11 D0 C1
0081:09 10 0D 11 0D 4C 39 A4 2D
0089:93 0D 0D 20 20 1C 12 82 63
0091:00 20 2A 2A 2A 20 57 4F B6
0099:52 40 2D 53 43 52 45 45 35
00A1:4E 20 2A 2A 2A 02 00 20 D3
00A9:92 02 05 0D 02 04 28 1E F4
00B1:54 47 20 52 45 2D 45 4E 22
00B9:54 45 52 28 41 46 54 45 24
00C1:52 20 45 58 49 54 28 54 E2
00C9:4F 20 42 41 53 49 43 0D A4
00D1:00 82 04 28 54 59 58 45 DA
00D9:20 12 20 47 45 4E 49 45 B3
00E1:20 92 20 41 4E 4A 28 50 54
00E9:52 45 53 53 20 52 45 54 B8
00F1:55 52 40 8D 02 04 28 F5
00F9:06 20 4F 52 28 12 20 32
0081:53 59 53 28 35 38 38 0B 0B
00B9:30 20 92 28 29 02 06 0D 7A
00C1:02 84 20 81 58 52 45 53 6B
00C9:53 20 12 53 48 49 46 54 1D
00D1:20 43 52 92 28 54 4F D6
00D9:20 43 45 41 52 28 53 61
00E1:43 52 45 45 4E 02 07 0D 8C
00E9:02 04 20 1F 43 55 52 52 A4
00F1:45 4E 54 28 43 4F 4C 4F D5
00F9:52 49 4E 44 49 43 41 2A 2A
0081:54 45 44 28 48 45 52 45 2B
0089:2D 20 2D 2D 2D 3C 10 00 C8
0091:90 13 1D 1D 43 4F 4D 5C B8
0099:55 54 45 21 1D 58 55 42 8D
00A1:4C 49 43 41 54 49 4F 4E AD
00A9:53 10 50 52 45 53 45 4E 7C
00B1:54 53 82 44 11 02 08 9D D2
00B9:1D 1D 1D 1D 1D 1D 1D 1D
00C1:1D 1D 1D 1D 1D 1D 1D 1D
00C9:75 02 26 63 69 62 28 90
00D1:75 63 69 75 63 69 06 63 D2
00D9:69 75 63 69 75 63 69 B2 11
00E1:20 B2 28 75 63 69 75 89
00E9:63 69 32 32 32 32 B2 75 A2
00F1:63 69 32 32 32 32 28 20 64
00F9:6A 63 69 62 28 62 63 68
0081:63 A8 63 28 B3 28 62 4D
0089:60 62 28 62 63 62 62 4D
0091:B3 28 62 62 62 62 A8 7E
0099:B3 28 28 62 62 62 28 62
00A1:6A 6B 6A 63 6B 61 20 33
00A9:60 6A 63 6A 63 6B 61 6B
00B1:20 B1 28 6A 63 6B 6A 3A
00B9:63 6B 20 B1 20 B1 20 B1 3E
00C1:63 6B 20 62 6A 02 26 1F
00C9:63 6B 99 75 62 12 68 69
00D1:9C 75 82 12 68 69 99 62 81
00D9:12 43 4F 4D 4D 41 4E 44 A6
00E1:53 3A 92 28 20 43 54 52 B5
00E9:4C 28 26 62 9C 62 12 1F
00F1:20 43 55 52 53 4F 52 28 23
00F9:46 55 4E 43 54 49 4F 4E 09
0081:53 28 32 62 69 62 02 12 7C
0089:20 62 9C 62 02 12 38 62 6B
0091:99 62 48 28 48 45 4C 58 77
0099:28 28 48 48 49 53 28 4D 4C
00A1:45 4E 55 29 62 9C 62 4E 8A
00A9:31 2D 44 52 41 57 82 8A 15
00B1:20 5F 62 99 62 42 2D 43 9E
00B9:4C 45 52 20 54 4F 49 28 F4
00C1:42 4F 54 54 4D 40 62 3E 38
00C9:9C 62 46 54 4F 49 49 B8
00D1:43 48 41 4E 47 45 20 43 8A
00D9:41 53 45 28 28 20 5F 62 22
00E1:99 62 2D 4D 4F 56 45 E2
00E9:02 8C 2D 62 9C 62 46 37 CF

```

0061:1	20	52	45	56	45	52	53	45	10	0059:91	10	60	24	0C	10	00	40	70	1071:0F	A9	00	85	10	A9	FC	85	82
0069:02	07	28	57	52	99	62	5A	04	0061:00	B1	0E	49	40	91	0E	60	A3	1079:11	20	87	A3	A9	00	85	0E	C7	
0071:20	40	45	40	47	52	49	5A	9F	0069:24	0C	50	0A	00	00	05	8A	2E	1081:A9	F8	85	0F	15	05	50	10	DF	
0079:45	20	53	43	52	45	45	4E	59	008A:91	0E	A5	00	10	10	60	18	74	1089:A5	F8	51	05	11	20	E7	A3	24	
0081:20	62	9C	6A	82	12	60	FE	0089:20	92	A4	20	30	A6	A9	27	07	C7	1091:10	00	05	0E	A9	FC	85	0F	A5	
0089:99	62	47	2D	4F	50	53	60	0091:11	85	10	A9	C0	85	11	20	08	1F	1099:52	05	10	15	53	53	11	20	93	
0091:21	20	28	52	45	53	54	4F	09	0091:A6	4C	00	A0	1B	20	92	A4	00	10A1:17	A0	29	20	8D	15	D0	A9	E1	
0099:52	45	29	20	62	9A	75	02	00	0091:29	30	A6	A5	0E	05	10	A5	04	10A9:A3	BD	F8	C8	4C	00	A0	A9	64	
00A1:12	60	69	99	62	50	20	43	91	0091:0F	05	11	A9	00	05	0E	A0	C6	10B1:10	00	03	35	05	3A	60	D2	19	
00A9:52	45	41	54	45	20	58	52	45	0091:30	05	0F	20	8B	A6	4C	00	6E	10B9:FF	20	C5	A9	43	40	20	00	A8	
00B1:49	48	54	28	50	47	62	29	00	0091:0A	EE	20	D0	4C	00	A0	EE	05	10C1:60	C9	01	D0	06	06	29	08	CE	
00B9:9A	62	12	06	20	43	47	9E	00	00A1:21	D0	4C	00	A0	A5	0D	49	AA	10C9:ED	FE	00	E9	11	D0	06	E4	54	
00C1:4C	47	52	02	07	28	92	62	7C	00A9:00	85	0D	20	30	A6	20	DC	B3	10D1:3B	D0	03	F8	E4	C9	9D	D0	A9	
00C9:99	62	58	20	45	58	49	54	95	00B1:0A	4C	00	A0	A5	0D	49	0F	00	10D9:06	CA	3A	38	D0	F0	CA	94		
00D1:20	54	20	42	41	53	49	FF	00	00B9:85	0D	20	30	A6	20	DC	B3	58	10E1:1D	D0	06	06	3A	3C	00	CA	F0	
00D9:43	20	20	62	9A	62	02	83	00	00E1:4C	00	A0	10	20	92	A4	4C	1A	10E9:20	D0	60	4A	32	D0	B1	AA	A4	
00E1:12	20	62	99	6A	02	12	60	CE	00E9:00	A0	A0	11	D0	49	40	8D	2E	10F1:3E	B1	0E	91	10	A9	20	91	71	
00E9:6B	9A	62	46	32	2D	42	4F	27	00F1:11	D0	29	40	F0	00	A0	16	9C	10F9:0E	08	18	F5	20	87	AA	A5	BE	
00F1:52	44	45	52	02	09	20	62	B6	00F9:D0	29	EF	8D	16	D0	4C	00	A0	1101:0E	18	69	28	05	0E	90	02	67	
00F9:95	75	02	12	60	69	9A	62	5A	00B1:0A	4C	00	A0	16	D0	49	10	D0	1109:0E	0F	05	A5	18	18	69	28	54	
00C0:46	34	20	42	41	43	40	47	00	00B9:D9	29	10	F0	00	A0	11	D0	53	1111:10	90	02	E6	11	CA	10	D4	B6	
00C9:52	4F	55	48	44	02	05	20	82	00C1:29	BF	0D	11	D0	4C	00	A0	4D	1119:60	18	20	ED	AD	4C	00	00	A8	
00C1:62	96	62	12	20	40	4F	44	06	00C9:AD	10	D0	49	84	29	FC	8D	8D	1221:20	78	A0	38	20	92	20	1F	1F	
00C9:45	53	3A	20	92	20	43	5A	21	00F1:10	D0	4C	00	A0	A5	0C	49	84	1129:C9	AA	A9	00	A0	C8	A2	04	00	
00C1:52	40	20	26	20	20	20	62	6F	00A9:00	85	0C	20	30	A6	20	E0	30	1131:20	EF	AA	A9	8C	A0	D0	A2	0C	
00C9:9A	62	46	36	20	42	41	43	8B	00B1:0A	4C	00	A0	A5	0C	49	40	60	1139:0A	20	EF	AA	A5	0C	20	8C	7E	
00C3:4B	47	4E	44	20	52	45	47	EA	00B9:05	0C	20	30	A6	20	C3	A1	81	1141:0A	AA	2A	8D	B6	D9	8A	0A	6D	
00C9:20	31	20	33	62	96	62	02	8A	00C1:20	F5	A0	4C	00	A0	A0	00	00	1149:2A	0D	66	D9	A5	0D	29	C0	C2	
00C1:12	20	62	9A	62	46	38	20	2A	00C9:31	0E	05	8A	B1	10	05	05	77	1151:8A	0D	0A	8D	B6	D9	8A	0A	3C	
00C9:43	48	41	52	53	20	47	46	62	00D1:60	30	20	92	A4	20	30	A6	06	1159:2A	0D	00	D9	A5	40	29	40	4E	
00C5:12	53	41	4D	45	20	20	62	98	00E9:A0	00	B1	0E	29	FF	2C	11	0F	1161:8A	0D	0A	8D	32	D0	01	47	BC	
00C9:96	62	48	20	45	50	54	45	08	00E1:00	50	02	29	3F	C9	20	08	7F	1169:29	18	F0	82	A9	01	8D	5A	29	
00C1:4E	44	45	44	20	42	41	43	6E	00F1:03	4C	00	A0	B1	10	20	0F	D3	1171:D0	A5	46	29	04	05	4C	D0	B1	
00C9:4B	47	44	5F	62	9A	62	20	D9	00F1:85	A4	A9	00	85	0E	05	10	10	1179:02	D0	A9	00	85	05	20	DC	6D	
00C7:20	20	43	4F	4C	4F	52	20	63	00F9:A0	D0	05	11	A9	FC	05	0F	3E	1181:A5	24	55	30	37	30	20	ED	74	
00C9:41	53	20	43	48	41	52	20	40	00F1:A0	00	A2	04	20	B1	AA	B1	10	1189:AA	4C	00	A0	48	38	20	D0	6C	
00C1:20	62	96	62	43	2D	40	55	9F	00F9:0E	20	87	A4	20	0F	C5	4A	2D	1191:4A	60	4C	09	A0	00	00	D0	C5	
00C9:4C	54	49	43	4F	4C	4F	52	D6	00F1:00	09	B1	10	10	69	0F	29	A2	1199:00	05	0E	A9	AF	05	0F	A9	19	
00C1:02	05	20	5F	62	9A	62	20	48	00F9:0F	91	10	C0	D0	E6	E6	11	82	11A1:00	40	D0	00	05	A9	41	B5	E6	64
00C9:20	20	47	56	45	52	20	43	10	00F1:86	0F	CA	D0	DF	20	C3	A5	99	11A9:A9	00	05	0F	A9	F8	48	A0	AA	
00C1:55	53	41	4F	52	02	04	20	18	00F9:39	0C	F0	C6	4C	09	A0	3D	99	11B1:07	AD	11	D0	91	0E	88	AD	75	
00C9:62	96	62	55	20	55	53	45	84	00F1:20	92	A4	20	CD	AA	A9	A0	00	11B9:16	D0	91	0E	88	AD	10	D0	2A	
00C1:52	20	44	45	46	49	4E	45	11	00F9:AD	A2	AE	20	9B	A5	A9	31	85	11C1:91	0E	88	89	28	D0	29	0F	82	
00C9:4A	20	20	5F	62	9A	6A	26	0F	00F1:10	D0	F9	AE	A9	D6	A2	AE	91	11C9:91	0E	88	18	F6	AD	06	02	C7	
00C1:02	12	60	60	62	20	20	00	00	00F9:28	9B	A5	B6	10	A5	10	C9	B2	11D1:05	A9	A5	C7	85	40	28	00	D0	
00C9:43	48	41	52	28	53	45	54	18	00F1:14	D0	EE	A9	00	A2	AE	20	A0	11D9:A0	06	29	84	2A	68	A9	A9	CE	
00C1:20	40	20	36	31	34	34	30	64	00F9:9B	A5	A9	1C	BD	15	D0	A0	A0	11E1:C8	A2	04	28	EF	AA	A4	11	10	
00C9:62	90	28	50	52	45	53	F8	00	00F1:02	B9	43	0E	05	4C	A5	0F	D2	11E9:A9	D0	A2	84	28	EF	AA	06	22	
00C1:53	20	41	4E	59	20	40	45	0F	00F9:18	F7	20	F3	A9	C9	20	D0	56	11F1:00	00	11	38	28	92	A4	A9	20	
00C9:59	0D	96	6A	82	12	60	60	EF	00F1:0C	A9	00	8D	15	D0	38	20	55	11F9:08	05	0E	A9	AF	05	0F	A9	79	
00C1:98	20	28	46	4F	52	20	57	22	00F9:ED	AA	4C	00	00	C9	31	90	60	1201:00	48	D0	1C	A9	41	85	0E	06	
00C9:47	52	48	2D	53	43	52	45	1C	00F1:29	C9	34	00	E5	38	20	38	AC	1211:40	05	C7	A5	49	AD	06	02	1F	
00D1:45	48	00	AD	06	02	8D	2C	AF	00F9:A0	06	A2	EE	8A	99	42	00	0F	1219:A6	29	AA	24	28	D0	2F	25	35	
00D9:00	20	3C	A5	24	55	10	19	16	00F1:4C	5C	A2	18	20	92	AA	20	77	1221:A0	07	00	00	8D	11	D0	00	00	
00D1:A0	12	D9	37	AF	F0	05	06	02	00F9:AC	A3	A9	18	85	30	A9	27	41	1221:0F	01	0E	16	D0	00	01	0E	D6	
00D1:10	F8	38	0D	90	AA	A8	89	48	00F1:A5	3C	A9	1E	8D	F0	C8	20	A0	1231:00	10	D0	88	B1	0E	99	20	D0	
00D1:4A	AF	48	C8	89	4A	AF	48	00	00F9:B6	A3	06	29	06	39	8A	2A	88	1239:00	00	10	D0	F0	20	B1	0E	AA	76
00D9:60	C9	94	F0	D6	C9	14	D0	5E	00F1:84	3A	A5	0E	05	4C	A5	0F	D2	1241:C8	A2	04	68	28	EF	AA	A5	35	
00D1:0A	95	A2	AC	20	90	A5	4C	52	00F9:05	AD	A5	10	85	4E	A5	11	65	1249:0F	AF	D0	A2	04	20	EF	AA	D0	
00D9:4C	00	A0	C9	22	F0	CA	C9	52	00F1:85	4F	AD	00	8D	8D	0C	D0	00	1251:28	87	AA	20	80	1A	20	81	EF	
00D1:93	40	28	08	AB	68	86	29	F8	00F9:AD	01	D0	8D	8D	0D	AD	10	89	1259:AA	A9	F0	A0	00	A2	00	20	A6	
00D1:0A																											

12F9:80	85	55	A8	00	04	12	A9	14	1581:90	08	C9	48	90	0A	C9	60	D3	1809:00	F2	60	38	28	F8	FF	C0	73
1301:FE	2D	15	D0	8D	15	D0	8A	8E	1589:90	03	18	69	28	18	69	20	AA	1811:28	90	04	98	E9	28	AB	60	25
1309:160	18	20	92	AA	A2	12	8D	44	1591:20	45	AA	A5	24	C5	25	F0	7A	1819:20	53	AB	A9	55	A2	AE	20	F0
1311:90	AF	95	08	1A	F8	A2	C9		1599:16	A5	54	C9	47	90	9C	28	35	1821:98	A5	28	F9	A9	48	2E	20	39
1319:22	8D	83	AF	95	39	1A	08	68	15A1:AA	AA	20	0E	AA	28	BF	AA		1829:08	AA	20	6C	AA	A5	43	20	68
1321:F8	78	D2	A6	AF	18	20	87		15A9:AF	08	85	54	4C	38	AF	20	53	1831:5B	AA	20	6C	AA	AE	AB	AD	E2
1329:80	FF	AF	04	8D	08	02	58	76	15B1:9A	08	C6	26	F0	1C	A5	24	76	1839:A5	44	28	5B	AA	8D	86	AD	42
1331:4C	76	C6	A9	08	85	8F	85	48	15B9:C9	27	D0	86	20	AA	4C	5A		1841:8E	AD	85	A5	45	28	58	AA	AD
1339:18	38	20	F0	FF	98	C9	28	AE	15C1:0E	AB	A9	22	28	2E	AA	A9	2D	1849:80	C8	AD	8E	8F	AD	A9	AA	88
1341:90	83	E9	28	AA	0A	0A	88		15C9:28	28	2E	AA	20	6C	AA	4C	C0	1851:A2	D0	20	41	AA	68	A9	18	98
1349:26	10	69	18	BD	08	D0	90	D3	15D1:0A	AB	20	AA	20	6C	AA	33		1859:A2	AE	28	98	A5	68	A9	91	07
1351:02	85	10	AD	10	20	29	FE	24	15D9:A5	28	85	1A	A5	21	85	18	89	1861:A2	AE	20	98	A5	A9	26	AD	0F
1359:05	10	8D	10	D0	0A	0A	8A	87	15E1:A9	22	A2	AD	20	98	A5	20	73	1869:A2	20	98	A5	20	F9	A9	68	63
1361:0A	85	10	69	32	8D	01	D0	21	15E9:AF	A9	D0	46	A5	27	85	14	E2	1871:A9	26	A2	AE	20	98	A5	28	1A
1369:A5	18	0A	26	0F	AA	26	0F	86	15F1:A5	28	85	15	20	8E	AA	20	A9	1879:AF	A9	60	A2	22	85	08	9D	DC
1371:18	65	10	8E	02	86	8F	84	56	15F9:6C	AA	A9	96	A2	AD	20	41	F0	1881:98	AF	CA	10	F8	AD	22	85	0C
1379:10	18	65	10	85	0E	85	10	DA	1601:AA	A5	47	29	10	F0	15	20	F5	1889:39	9D	83	AF	CA	10	F8	AD	22
1381:A9	88	65	0F	85	8F	69	10	97	1609:5B	AB	D0	26	18	0E	AA	20	F8	1891:D6	AF	38	28	D0	FF	78	7E	5A
1389:85	11	60	80	E9	A9	20	91	45	1611:6C	AA	A9	E9	A2	AD	20	41	3F	1899:28	8A	FF	A9	EF	D0	18	03	FF
1391:8E	A6	0E	84	10	D8	86	AE	EE	1619:AA	4C	2F	A9	24	48	50	12	23	18A1:AF	C6	D0	19	83	A9	9A	8D	1D
1399:0F	E4	11	F0	08	86	0E	D8	7A	1621:20	58	AB	D0	8D	20	0E	AA	66	18A9:14	83	A9	8D	0E	15	83	58	65
13A1:EE	E6	0F	D0	EA	6F	A9	28	34	1629:28	6C	AA	A9	C2	A2	AD	20	CB	1881:A9	08	D0	27	D0	85	0D	85	17
13A9:8D	AF	C8	38	28	92	AA	20	42	1631:41	AA	A5	3F	F0	1F	A9	36	D2	1889:12	85	0C	0D	17	D0	AD	01	87
13AB:CD	AA	A9	93	20	D2	FF	A9	CD	1639:A2	AD	20	98	A5	A6	1C	A5	85	18CL:03	C9	C6	F0	13	D0	C2	C6	E8
1389:08	85	14	85	15	85	40	A9	05	1641:1D	28	41	AC	A9	53	A2	AD	84	18C9:AD	08	83	AD	1C	C5	A9	8C	43
13C1:FF	A2	AC	28	98	A5	A9	15	04	1649:28	98	A5	A6	27	A5	28	20	CB	18D1:8D	08	A9	C6	8D	81	83	35	
13C9:A2	AF	20	98	A5	28	C3	A5	C5	1651:41	AC	4C	84	A6	A5	27	D0	09	18D9:AF	84	AB	C8	A2	84	28	EF	D5
13D1:4C	8E	86	38	28	8D	AA	4C	FE	1659:02	C6	28	C6	27	A5	28	C9	93	18E1:AA	AD	82	D0	83	8D	02	62	
13D9:80	A8	A9	86	A2	AC	28	98	6C	1661:FA	98	86	28	85	AA	4C	35	18	18E9:D0	A9	08	8D	8D	AD	18	07	
13E1:A5	4C	84	26	28	C3	A5	C9	14	1669:A7	A5	1A	38	E9	00	85	10	84	18F1:D0	29	82	09	28	8D	1C	D0	E7
13E9:D0	F0	43	C9	20	F0	E4	C9	30	1671:A5	18	E9	D0	85	11	A5	2D	6A	18F9:AF	C9	8D	88	02	A9	1C	AD	1E
13F1:30	90	E7	C9	3A	88	83	28	6A	1679:85	0E	18	65	10	85	10	A8	AA	1981:F8	C8	88	04	A9	1D	8D	FE	EC
13F9:D2	FF	29	0F	AF	08	14	A6	4A	1681:A5	2E	85	0F	65	11	85	11	39	1989:CB	88	08	FA	A9	1F	8D	FE	EC
1401:15	86	14	26	15	88	D3	86	33	1689:C5	38	90	1A	D0	84	C4	37	81	1911:CB	A9	1F	8D	FE	EC	AD	87	C7
1409:14	26	15	88	C0	65	14	85	24	1691:90	14	A9	57	A2	AD	20	98	5D	1919:89	88	AF	99	84	D0	88	18	50
1411:14	8A	65	15	88	C4	85	15	9C	1699:A5	A9	15	A2	AF	28	98	A5	AA	1921:F7	AD	D0	8D	1B	D0	A9	1C	AD
1419:06	14	26	15	88	C0	98	65	6E	16A1:20	93	4C	D8	6A	A2	85	40		1929:88	1D	D0	8D	1B	D0	A9	1C	AD
1421:14	85	14	A5	15	69	08	88	98	16A9:95	2D	CA	94	20	CA	10	F0	3C	1931:28	D2	FF	68	78	48	A9	24	8A
1429:81	85	15	4C	K1	A6	A5	14	F8	16B1:20	73	A5	A5	1A	85	0E	A5	82	1939:85	81	68	68	A9	26	85	3C	
1431:85	1C	A5	15	C9	FA	90	87	8B	1689:1B	85	8F	A9	08	85	5F	A9	CF	1941:87	68	58	68	06	56	85	57	0F
1439:A9	DC	A2	AC	4C	D8	86	85	31	16C1:D0	85	68	20	73	A5	28	7C	84	1949:A2	83	A9	08	95	58	CA	10	76
1441:1D	A8	80	85	1A	A9	D8	85	59	16C9:C6	28	85	AA	4C	96	A9	28	F2	1951:F8	A0	8F	86	56	27	78	5E	8A
1449:18	A5	47	29	18	F0	1D	28	82	16D1:81	AA	81	16	28	87	AA	60	D5	1959:F8	A5	58	65	58	85	58	A5	82
1451:15	AB	20	53	AB	28	6D	AB	8C	16D9:28	81	AA	81	18	28	87	AA	E1	1961:59	65	59	85	59	A5	5A	65	98
1459:20	38	A9	08	8D	87	AD	20	F1	16E1:29	8F	08	A9	19	85	26	A9	E1	1969:A5	85	5A	D8	58	88	18	E3	98
1461:0E	AA	A9	D6	A2	AD	20	41	28	16E9:85	85	16	85	18	A9	F0	85	71	1971:A2	82	85	58	48	4A	4A	4A	F8
1469:AA	4C	87	A7	24	48	58	1A	62	16F1:17	A9	FC	85	19	68	28	E4	78	1979:A2	28	8A	CA	28	29	8F	28	1B
1471:20	15	AB	20	53	AB	28	6D	5D	16F9:F8	FF	F8	68	20	E4	FF	F0	6D	1981:8A	CA	10	ED	A5	58	8D	0C	8C
1479:A9	D8	0F	A9	3A	8D	87	AD	20	1701:F8	C9	59	F0	88	C9	0D	F0	64	1989:03	20	8E	AC	68	C5	58	F0	A3
1481:28	8E	AA	A9	FD	A2	AD	20	23	1709:87	08	A9	4E	20	D2	FF	28	4B	1991:F8	A9	30	85	5B	4C	D2	FF	14
1489:14	AA	20	8E	AA	A5	41	20	78	1711:60	A9	FF	20	2E	AA	28	2E	66	1999:28	9D	9D	80	93	1F	82	0A	1E
1491:58	AA	8D	7E	AD	8E	7D	AD	F8	1719:AA	A5	14	20	2E	AA	A5	15	87	19A1:8D	4C	41	53	54	20	4C	49	CP
1499:A5	42	28	5B	AA	8D	80	AD	29	1721:20	2E	AA	20	85	C6	90	02	AC	19A9:4E	A5	20	48	55	4D	42	45	86
14A1:8E	87	AD	A5	46	29	02	F0	8C	1729:8E	AD	86	14	D0	82	E6	15	6A	19B1:52	28	55	53	45	44	3A	28	C4
14A9:8A	A9	FF	D0	82	A9	32	8D	F3	1731:68	28	31	AC	80	91	1A	CA		19B9:08	93	12	1C	2A	2A	2A	20	43
14B1:8F	AD	20	53	AB	A9	72	A2	D1	1739:28	39	AC	86	1A	D0	82	E6	C8	19C1:49	4E	56	41	4C	49	44	20	38
14B9:A2	28	9B	A5	28	F9	A9	D0	1C	1741:18	86	54	68	85	18	86	11	D1	19C9:45	4E	54	52	59	28	2A	2A	20
14C1:87	A9	76	A2	AD	4C	C9	A7	AA	1749:AD	08	81	18	C9	FF	F0	83	32	19D1:2A	92	28	54	52	59	28	21	88
14C9:A9	8A	A2	AD	20	41	AA	28	21	1751:28	2E	AA	86	18	D0	82	E6	8D	19D9:47	41	49	A4	4E	28	8D	0D	8C
14D1:6C	AA	24	48	78	26	20	E8	14	1759:11	C9	08	D8	88	6D	A2	FF	85	19E1:1C	4C	49	4E	A5	28	4E	55	DF
14D9:A9	AD	27	28	CC	A9	C9	68	8E	1761:29	8F	C9	0A	98	86	A2	31	D4	19E9:40	42	A5	52	28	A5	58	43	2B
14E1:F8	8F	C9	AD	F0	84	C9	88	95	1769:18	69	26	68	69	38	68	A5	38	19F1:45	A5	44	53	28	36	33	39	47
14E9:D0	8B	20	D5	A																						

1A99:00	97	31	39	38	2C	38	3A	7A
1AA1:92	31	39	38	2C	31	00	97	D3
1AA9:35	33	32	38	32	2C	38	38	E1
1AB1:3A	97	35	33	32	38	33	2C	C7
1AB9:30	38	3A	97	35	33	32	38	85
1AC1:34	2C	38	38	00	97	35	33	20
1AC9:32	36	35	2C	C2	28	35	33	62
1AD1:32	36	35	29	AF	31	39	31	CB
1AD9:00	97	35	33	32	37	38	2C	CB
1AE1:C2	28	35	33	32	37	38	29	53
1AE9:B0	31	36	00	97	35	33	32	83
1AF1:37	38	2C	C2	28	35	33	32	28
1AF9:37	38	29	AF	32	33	39	00	C6
1B01:97	35	33	32	36	35	2C	C2	78
1B09:28	35	33	32	36	35	29	00	B3
1B11:36	34	00	0D	0D	41	44	44	7A
1B19:28	4C	49	48	45	20	54	4F	23
1B21:28	53	45	54	20	58	0D	0D	28
1B29:00	45	58	54	45	48	44	45	32
1B31:44	28	42	41	43	4B	47	52	16
1B39:4F	55	42	44	28	4D	4F	44	93
1B41:45	28	47	52	0D	4D	55	4C	C5
1B49:54	49	43	4F	4C	52	28	4D	92
1B51:4F	44	45	3F	28	59	9D	00	7E
1B59:42	41	43	48	47	52	4F	55	95
1B61:4E	44	28	52	45	47	49	53	26
1B69:54	45	52	53	20	31	2D	33	ED
1B71:1F	28	59	9D	00	42	41	43	23
1B79:40	47	52	4F	55	4E	44	28	F2
1B81:26	28	42	4F	52	44	45	52	90
1B89:20	43	4F	4C	4F	52	53	3F	F8
1B91:28	59	9D	00	0D	0D	41	44	45
1B99:44	28	4C	49	4E	45	20	54	34
1BA1:4F	28	54	55	52	4E	28	4F	C2
1BA9:46	46	0D	00	93	0D	1C	12	51
1BB1:02	85	28	58	52	45	53	53	D4
1BB9:20	48	45	59	53	20	31	2D	BB
1BC1:33	28	54	4F	20	43	59	43	1D
1BC9:4C	45	20	43	4F	4C	4F	52	4C
1BD1:53	02	05	20	92	02	03	0D	04
1BD9:00	0D	97	02	1D	28	6F	02	00
1BE1:84	87	70	0D	02	05	28	42	BD
1BF9:41	43	48	47	52	4F	55	48	38
1BF1:44	28	52	45	47	49	53	54	48
1B99:45	52	20	20	38	28	28	45	55
1C01:02	84	20	A7	0D	02	1D	20	84
1C09:6C	02	04	AF	BA	0D	00	00	97
1C11:0D	0D	00	12	1F	02	06	20	03
1C19:50	52	45	53	53	28	27	53	AB
1C21:50	41	43	45	23	28	46	4F	24
1C29:52	28	57	47	52	48	2D	53	DF
1C31:43	52	45	45	48	02	05	28	41
1C39:92	00	89	8A	8B	8C	85	86	84
1C41:07	08	02	14	0D	0F	08	18	A5
1C49:18	1A	08	03	15	A1	35	A1	08
1C51:3A	02	08	A1	CD	A1	88	A1	47
1C59:41	A1	A1	A1	58	A1	03	A1	99
1C61:18	42	3F	A4	15	A4	1F	A6	AA
1C69:42	A6	05	A1	5F	A1	66	A1	47
1C71:7D	A1	94	90	05	1C	9F	9C	08
1C79:1E	1F	9E	81	95	96	97	98	43
1C81:99	9A	98	08	63	08	08	08	D2
1C89:83	58	F2	20	8F	C6	4C	1D	55
1C91:A4	28	93	C6	4C	69	FE	28	19
1C99:93	C6	28	33	A5	4C	8E	C6	C6
1CA1:28	93	C6	28	13	A6	4C	8E	04
1CA9:0E	A3	26	85	81	68	48	A9	98
1CB1:27	85	81	68	68	A5	12	FE	34
1CB9:1B	C6	13	0D	17	A9	14	85	AD
1CC1:13	AD	27	D8	49	81	8D	27	71
1CC9:D0	AD	2D	D8	49	81	8D	2D	1F
1CD1:D0	8D	28	4D	4C	31	KA	08	05
1CD9:0B	F8	03	4C	8B	23	A5	7A	AA
1CE1:0D	08	C6	A5	7B	8D	DC	C6	9D
1CE9:A0	84	AD	DB	C6	08	03	C6	35
1CF1:DC	C6	CE	DB	C6	AD	01	08	08
1CF9:D9	EA	C6	D8	08	18	EA	EA	E3
1D01:68	68	4C	78	C6	47	45	4E	46
1D09:49	45	48	AA	08	08	08	08	EB

Screen Saver 64

Stephen E. Maslars

Here are two fast, useful routines for storing and retrieving high-resolution graphics screens with a disk drive. They work with the Commodore 64 or Commodore 128 in 64 mode.

Taking advantage of the Commodore 64's high-resolution graphics can be a time-consuming process at best. Even with extra commands such as those found in Simons' BASIC, it may take many minutes or even hours to plot a detailed screen. Utilities for dumping a high-resolution screen to your dot-matrix printer are readily available. At times, however, you may wish to save your graphics screen in a disk file so you can display it later without rerunning the program that created it. And if you own an Okimate 10 or similar color printer, the ability to save multicolor graphics screens is particularly useful.

"Screen Saver 64" provides two machine language (ML) routines that let you quickly save and retrieve hi-res graphics screens—both standard and multicolor—

from disk. Though they're written in ML, you can use them without knowing the ins and outs of ML yourself. And we've included two demonstration programs that show exactly how to use the ML routines for real applications.

To get started, type in and save Programs 1 and 2. Program 1 puts the screen save routine into memory, and Program 2 creates the screen retrieval routine. Since both ML routines go into the same memory area, they must be used separately. If you have an ML monitor and wish to examine the routines, note that each is broken into two parts, located from memory locations 679-738 and 828-1023 (decimal).

Saving A Graphics Screen

Here are the steps for saving a graphics screen with Screen Saver 64:

1. Run Program 1 to place the ML screen save routine in memory.
2. Create your hi-res or multicolor screen as usual. If you don't know how to do this, the Commodore 64

Programmer's Reference Guide and many other books explain the required steps. Program 3 (see below) contains a simple demonstration.

3. Execute a statement like OPEN 2,8,2,"filename,P,W" to open a disk file for writing (replace filename with the name of your own file). You must open the file as a PRG (program format) file using the ,P suffix as shown above. The ,W suffix indicates that you're opening the file for a write operation, and the first numeral 2 sets the logical file number (2 in this case) for that file.

4. Execute SYS 1007 to activate the ML save routine. *This must be done while you are in hi-res or multicolor mode.* The ML routine finds the currently defined graphics screen and associated color memory, and stores their contents in the disk file.

5. Execute a statement like CLOSE 2 to close the file. It is very important that you end the procedure by CLOSING the file, specifying the same logical file number (2 in this case) which was used to open it. If you omit this vital step, you may end up with a poison (unclosed) file on the disk that could damage other files or render the whole disk unreadable.

Retrieving A Graphics Screen

Once you have saved the screen to disk, it's easy to retrieve. Here are the steps to follow for bringing a graphics screen back into memory:

1. Run Program 2 to put the ML retrieval routine in memory.

2. Perform the steps needed to enter the appropriate hi-res or multicolor graphics mode.

3. Execute a statement like OPEN 2,8,2,"filename,P,R" to open the disk file for reading (input). Again, the ,P suffix specifies a PRG file, and the ,R suffix opens the file for reading.

4. Execute SYS 881. The ML routine loads the graphics data back into the right memory locations.

5. Execute a statement like CLOSE 2 to close the disk file. Again, you should use the same logical file number (2 in this case) used when opening the file.

6. At this point you can continue

with a BASIC program or do whatever else you like.

Graphics Demonstrations

Programs 3 and 4 contain practical demonstrations of how to use these two routines from BASIC. Type in and save both programs, then load and run Program 1 to put the ML save routine in memory. Now load and run Program 3. This program uses the hi-res drawing example from pages 123-126 in the *Commodore 64 Programmer's Reference Guide*. Lines 110-140 define the hi-res screen and color memory to start at locations 8192 and 1024, respectively, then clear the graphics screen. Lines 150-230 draw a simple sine wave pattern. (Be patient; it takes a few minutes to complete the drawing.) Line 270 opens the disk file using 2 as the logical file number and SINEWAVE.HIRES as the filename. After checking the disk error channel, the program calls the ML save routine.

The sine wave disappears as the hi-res memory is moved temporarily to a new location and stored in the disk file. Then the routine moves the picture back to its original location, saves color memory, and returns control to BASIC. After checking the error channel again, the BASIC program restores the normal screen display and ends.

Program 4 shows how to use the ML retrieval routine. Since it looks for a file named SINEWAVE.HIRES on the disk, you can run it only after you've used Program 3 to create the file. Run Program 2 to put the ML retrieval routine in memory, then load and run Program 4. Lines 110-130 define the hi-res screen starting at location 24576, a different area than the one it was saved from. Lines 140-150 fill the screen with a uniform pattern. (Note that this is done only for the purpose of demonstration, to confirm that the retrieval routine puts new information on the screen. It is not necessary to clear the graphics screen before using this routine.)

Lines 160-190 open a disk file for reading, using the same name as Program 3 (SINEWAVE.HIRES). After checking the error channel (180-190), the retrieval routine is called with SYS 881. The hi-res

screen is restored right before your eyes: First the graphics information appears, then color memory is brought in as well. After a brief pause, the program restores the screen to normal and ends.

Inside The ML Routines

The ML save routine saves the currently defined graphics screen and its associated color memory wherever they are located—even if the hi-res screen is stored in the RAM underneath a ROM area. The ML retrieval routine brings the stored screen back into whatever area you have currently defined as the graphics screen, even if that's a different location from the area from which it was saved. This lets you create and store a complex graphics display using one particular graphics aid (Simons' BASIC, etc.) and retrieve it for use by any other program.

Since sprites are independent of other graphics, these routines can't store or retrieve sprite shapes that appear on the screen.

To make this routine compatible with as many programs as possible, memory usage is restricted to three areas. It uses memory locations 679-738 (normally unused) and 828-1023 (the cassette buffer) to store the routines, and also zero-page locations 2 and 251-254. To save a screen, the ML routine first looks in locations 56576 and 53272 to locate the graphics screen and normal screen memory (which becomes the hi-res color memory). It then swaps the 8K bytes of hi-res RAM memory with the contents of locations 24576-32767 (\$6000-\$7FFF). This is done by "turning off" the computer's ROM chips temporarily so the swapping routine can see hi-res memory no matter where it's located. Then the routine switches the Kernel ROM back in to write the 8,000 bytes of hi-res information to the disk file, and moves the hi-res screen back to its original location. The 1,000-byte screen memory area is written directly to the disk file.

Note that since no memory swapping is done for color memory, this part of your screen must be located in a memory area that's not normally hidden by ROM.

Finally, the normal color mem-

ory at 55296-56295 (used in multi-color mode) as well as the screen background byte at 53281 is written to disk. The final disk file is 10,003 bytes (40 blocks) long. Two extra bytes are added at the beginning of the file to make it compatible with version 3.0 of the Okimate Color Print program.

The retrieval routine works in reverse, finding where the graphics screen and color memory are located in the current configuration, then restoring everything to the correct memory locations. Since RAM can be POKED even if it's under ROM, no memory swapping is required and the contents of the disk file are moved directly into the appropriate memory areas.

For instructions on entering these listings, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE!

Program 1: Screen Saver 64

```
CS 100 REM PROGRAM 1 SCREEN SA
VER 64
PE 110 SU=0:FOR I=680TO739:READ
X:SU=SU+X:POKE I,X:NEXT
CK 120 IF SU<4055 THEN PRINT "ERR
OR IN DATA IN LINES 170
-220":GOTO 1670
QK 130 SU=0:FOR I=680TO1022:REA
D X:SU=SU+X:POKE I,X:NEXT
PQ 140 IF SU<31598 THEN PRINT "ER
ROR IN DATA IN LINES 24
0-470":GOTO 1670
SR 150 PRINT "SCREEN SAVE INSTA
LLED":END
MS 170 DATA173,0,221,41,3,73,3
,10,10,10
QP 180 DATA10,10,10,133,254,17
3,24,208
QJ 190 DATA41,8,10,10,101,254,
133,2,105
DA 200 DATA31,141,169,2,173,24
,208,41,240
DB 210 DATA74,74,234,234,101,2
54,141,168
XM 220 DATA2,105,3,141,167,2,9
6
KB 240 DATA160,0,132,251,132,2
53,165,2,133
PH 250 DATA252,169,96,133,254,
128,165,1,72
MG 260 DATA41,253,133,1,177,25
1,170,177
DA 270 DATA253,145,251,138,145
,253,208
RK 280 DATA208,243,230,252,230
,254,165
HK 290 DATA254,201,128,208,233
,104,133,1
JH 300 DATA88,234,234,234,96,1
62,2,32,201
MD 310 DATA255,169,0,32,210,25
5,165,2,32
SK 320 DATA219,255,160,0,132,2
51,169,96
RK 330 DATA133,252,234,234,177
,251,32,210
XS 340 DATA255,165,252,201,127
,240,7,208
```

```
PX 350 DATA208,242,230,252,200
,238,208
MX 360 DATA152,201,64,208,232,
96,160,0
FC 370 DATA132,251,173,168,2,1
33,252,177
FA 380 DATA251,32,210,255,165,
252,205,167
SH 390 DATA2,240,7,208,208,241
,230,252
MD 400 DATA208,237,208,152,201
,232,208
JB 410 DATA231,160,0,132,251,1
69,216,133
XS 420 DATA252,177,251,32,210,
255,165,252
JD 430 DATA201,219,240,7,208,2
00,242,230
XE 440 DATA252,200,230,208,152
,201,232
XC 450 DATA208,232,173,33,208,
32,210,255
PH 460 DATA32,204,255,96,32,17
6,2,32,60
MX 470 DATA3,32,113,3,32,60,3,
32,163,3,96
```

Program 2: Screen Retriever 64

```
PR 100 REM PROGRAM 2 SCREEN RE
TRIEVE
PE 110 SU=0:FOR I=680TO739:READ
X:SU=SU+X:POKE I,X:REX
HE 120 IF SU<4055 THEN PRINT "ERR
OR IN DATA IN LINES 160
-180":GOTO 1670
JQ 130 SU=0:FOR I=681TO1004:REA
D X:SU=SU+X:POKE I,X:REX
KK 140 IF SU<21298 THEN PRINT "ER
ROR IN DATA IN LINES 19
0-250":GOTO 1670
HM 150 PRINT "SCREEN RETRIEVE I
NSTALL":END
KB 160 DATA173,0,221,41,3,73,3
,10,10,10,10,10,133,
254,173,24,208
PD 170 DATA41,8,10,10,101,254,
133,2,105,31,141,169,2,
173,24,208,41,240
KP 180 DATA74,74,234,234,101,2
54,141,168,2,105,3,141,
167,2,96
PC 190 DATA32,176,2,162,2,32,1
98,255,32,207,255,32,20
7,255,234,160,0,132,251
AH 200 DATA165,2,133,252,32,20
7,255,145,251,165,252,2
05,169,2,240,7,208,208
QK 210 DATA241,230,252,208,237
,208,152,201,64,208,231
,160,0,132,251,173,160,
2
FR 220 DATA133,252,32,207,255,
145,251,165,252,205,167
,2,240,7,208,208,241,23
0
AB 230 DATA252,208,237,208,152
,201,232,208,231,160,0,
132,251,169,216,133,252
,32
JE 240 DATA207,255,145,251,165
,252,201,219,240,7,208,
208,242,230,252,208,238
,208
SH 250 DATA152,201,232,208,232
,32,207,255,141,33,208,
32,204,255,96,999
```

Program 3: Screen Saver Demo

```
DG 100 REM PROGRAM 3 SCREEN SA
VE DEMO
HS 110 BASE=24096:POKE53272,P
EEK(53272)OR0
NH 120 POKE53265,PEEK(53265)OR
32
SK 130 FOR I=BASE TO BASE+7999
:POKE I,0:NEXT
KD 140 FOR I=1024TO2023:POKE I,
3:NEXT
MS 150 FOR X=0 TO 319 STEP.5
RF 160 Y=INT(90*80*SIN(X/10))
KR 170 CH=INT(X/8)
QC 180 RO=INT(Y/8)
RF 190 LN=YAND7
AR 200 BY=BASE+RO*320+6*CH+LN
QC 210 SI=7-(XAND7)
BX 220 POKEBY,PEEK(BY)OR(2*BI)
AP 230 NEXT X
EG 240 POKE1024,16
GR 250 FOR I=1(2 SPACES)TO 100
0:NEXT(2 SPACES)I
BM 260 OPEN15,8,15
JK 270 OPEN2,0,2,"SINWAVE.HIR
ES,P,W"
RJ 280 INPUT15,Z1,Z2,Z3,Z4
AG 290 IF Z1<0 THEN CLOSE2:CL
OSE15:PRINT Z1:Z2:Z3:Z
4:GOTO350
MX 300 SYS 1007
AS 310 CLOSE2
AN 320 INPUT15,Z1,Z2,Z3,Z4
RS 330 IF Z1<0 THEN CLOSE15:P
RINT Z1:Z2:Z3:Z4
MH 340 FOR I=1 TO 1000:NEXT I
MM 350 POKE53265,PEEK(53265)AN
D 223
DF 360 POKE53272,PEEK(53272)
(2 SPACES)AND 247
DR 370 END
```

Program 4: Screen Retriever Demo

```
XF 100 REM PROGRAM 4 SCREEN RE
TRIEVE DEMO
PH 110 POKE56578,PEEK(56578)O
R 3:POKE 56576,(PEEK(56
576)AND252)OR 2
HD 120 BASE=64096:POKE53272,P
EEK(53272)OR0
FG 130 POKE53265,PEEK(53265)OR
32
FM 140 FOR I=BASE TO BASE+7999
:POKE I,66:NEXT
CS 150 FOR I=16384+1024 TO 163
84+2023:POKE I,77:NEXT
GE 160 OPEN15,8,15
QJ 170 OPEN2,0,2,"SINWAVE.HIR
ES,P,W"
CC 180 INPUT15,Z1,Z2,Z3,Z4
GG 190 IF Z1<0 THEN CLOSE2:CL
OSE15:PRINT Z1:Z2:Z3:Z
4:GOTO230
RS 200 SYS 881
PJ 210 CLOSE2
MC 220 FOR I=1 TO 2500:NEXT I
DF 230 POKE56576,(PEEK(56576)A
ND252)OR 3
FR 240 POKE53272,PEEK(53272)AN
D247
ND 250 POKE53265,PEEK(53265)AN
D223
```

Atari FontMaker

Charles Brannon, Program Editor

"FontMaker" simplifies the design of character sets for all text modes on Atari 400/800, XL, and XE computers. Although programmers will find FontMaker a valuable addition to their utility library, nonprogrammers can also benefit. Next month, we show how to use FontMaker to customize Atari SpeedScript's special character set. Whether you want Old English or a computer age font, FontMaker has the tools you need to express your creativity. It requires at least 16K RAM; disk drive recommended.

"FontMaker" is a sophisticated character editor written completely in machine language to work with all Atari text modes. Even though a character set (or font) editor is a handy utility for programmers, you don't need to know anything about machine language or programming to have fun with it. We include a simple subroutine that lets you load and merge character sets with your own BASIC programs. And next month, we'll show how to use the special ANTIC 3 character sets with the Atari version of our SpeedScript word processor (COMPUTE!, March 1986).

Since FontMaker is written in machine language for speed and compactness, you need to type it in with MLX, our machine language editor. See the MLX article elsewhere in this issue for instructions on typing in and using MLX to enter machine language programs.

When you run MLX, answer

the first three screen prompts like this:

Starting Address: 12288
Ending Address: 14867
Run/Init Address: 12288

Next you'll be asked "Tape or Disk?". Although FontMaker can load as a boot tape, it's much easier to use with a disk drive. If you press D for Disk, you'll be asked "Boot Disk or Binary File?". Press F to select binary file. FontMaker will run from a boot disk, but without DOS, there's no way to save or load character sets. So make sure you select F, since MLX can't convert from a boot disk to a binary file.

At the first screen prompt, 12288., start typing the data from Program 1. See the MLX article for a list of commands that let you type in a program in several sessions. When you've typed the last line, MLX prompts you for a disk filename. This will be the name under which FontMaker is saved to disk. If you're using Atari DOS 2.0S, 2.5, or 3.0, you may use the filename AUTORUN.SYS. This allows FontMaker to automatically load and run when you turn on the computer with the disk in the drive. Be sure this disk also contains the DOS files.

If you want to prevent loading FontMaker automatically (especially if you haven't finished typing it but want to save your preliminary typing), you can rename AUTORUN.SYS to some other name. You can also load FontMaker from DOS 2.0S or 2.5 with menu selection L. If you're using another DOS, such as OS/A+ or DOS XL, you may be able to save FontMaker with a different name, such as FONT.COM

and type the command FONT to run FontMaker.

Editing Characters

Assuming you've typed in and saved FontMaker, run it and follow along with this article. You'll see a screen with a colorful 8 x 8 grid, a list of brief instructions, and four rows of characters at the bottom of the screen (see screen photo).

When FontMaker starts, it asks you to pick a character. You can edit one character at a time. When you're asked to select a character, you can use the joystick to move a cursor around in the four rows of characters, then press the fire button to select the character highlighted by the cursor. Or you can simply press the keyboard key corresponding to the character. Action then shifts to the 8 x 8 character editing grid.

Within the grid, you can move the editing cursor (a hollow white box) with either the joystick or the cursor keys. You don't need to use CTRL with the cursor keys to move the cursor—CTRL-cursor up/down/left/right and SHIFT-cursor up/down/left/right are reserved for other features. When drawing in the character grid, press the fire button or the space bar to reverse (toggle) the dot at the cursor position. Previously set dots are turned off, and blank spaces are turned on. You can hold down the fire button while you move the joystick to draw lines and figures.

As you change the grid, you can see the character in actual size in the character set window. The cursor highlights (reverses) the selected character, but a row of the



"Atari FontMaker" lets you design your own custom character fonts for any Atari text mode.

character you're editing is also displayed. In addition, there's a sample of text ("The quick brown fox jumped over the lazy dogs") so you can judge relative character height and spacing.

To create an entirely new character, you may want to start by pressing SHIFT-CLEAR to erase the existing character pattern. This gives you a clean canvas for your design.

Undo Your Mistakes

If you don't like a change you've made, press U to undo all the changes made since you've selected the character. Press U again to undo the undo, restoring the change you've made. You can press F to fix a character, recopying its image from the standard character set stored in the computer's Read Only Memory (ROM). It's important to distinguish between these options: U reverts to the previous character image, F always gives you the ROM image. If you change the letter A to a spaceship, change B to a rocket, then go back to A and change the spaceship to an alien, U switches between the alien and the spaceship; F gives you the pattern for the letter A. Beware that you can't undo the Fix command.

If you hold down the OPTION button while pressing F, the entire character set is fixed, recopied from ROM. This wipes out any changes you have made to the character set, so be careful.

To select another character to edit, press P and use the joystick or keyboard to pick the new character. In some text modes (see the G command below), the joystick cursor may seem to move strangely. It

consistently moves up or down between rows of 32 characters. Since there are only 20 characters per line in modes 1 and 2, these rows wrap around the right margin. You move left and right within a row of characters, and up and down between rows of 32 characters.

FontMaker can display the character set in all Atari text modes. These are GRAPHICS 0 (the default text mode), "GRAPHICS 0½" (technically known as ANTIC 3, a nine-line true-descender mode), multicolor ANTIC mode 4, multicolor ANTIC mode 5, GRAPHICS 1, and GRAPHICS 2. Press G to cycle through these modes.

The SpeedScript Character Mode

If you've never heard of the ANTIC text modes, don't fret; they're not normally accessible from BASIC. The Atari SpeedScript word processor (and, incidentally, the PaperClip word processor from Batteries Included) uses the ANTIC 3 mode for large, readable characters. ANTIC 3, nicknamed GRAPHICS 0½, is a special 40-column mode that lets you define characters within an 8 X 10 character space. Other Atari text modes have only an 8 X 8 character grid. This means that ANTIC 3 characters can have true descenders. (A descender is the part of a character that drops below the line of type, such as the tail on a lowercase y or j.)

You still use just eight rows to define a character in ANTIC 3, but the character is positioned within ten screen scan lines. For uppercase characters, the eight rows of the character grid are displayed in rows 1-8 of the character matrix, with two blank lines at the bottom of each character, reserving space for lowercase descenders. For lowercase characters, the first two lines of the character grid are forced blank. The first two rows of the character definition are actually displayed at lines 9 and 10 of the character space, making it easy to reserve space for the descenders. The third through eighth rows of the character are displayed starting at the third line of the character space.

This may sound confusing, but fortunately FontMaker lets you design ANTIC 3 characters without

having to mentally translate what you see on the grid to what the character should look like. (However, the Rotate option seems to work strangely in this mode due to the unusual memory configuration used by ANTIC 3 characters.) Just be aware that lowercase characters are actually written two lines lower on the screen than uppercase characters. If you start with the normal character set, you'll need to use the roll and shift options to align the characters. Using the roll or shift commands (see below), roll the lowercase characters down two lines and all the uppercase characters down by one line. While you work, refer to the "quick brown fox" sentence to see that all the characters line up properly.

Since there are forced blank lines for uppercase characters, you can use the lower seven lines to define a large character. Leave the top line blank if you want two-line descenders. The normal Atari character set only uses six columns for a character, giving two pixel spaces between each character on the screen. A larger character can use up to seven columns, leaving the last column blank to keep characters from running into each other. When designing some character sets, though, such as a cursive script, you may want characters to connect together, so you can use the full horizontal space.

You can press the CTRL-cursor keys to roll the character within the grid. Pixels that are pushed off the edge of the grid wrap around to the opposite side. This is especially useful for those ANTIC 3 characters. If you press SHIFT with the cursor keys, the pixels that are shifted off the edge of the grid are lost. This can be used to crop a character, or quickly erase a certain column or row. Just roll the character until the column or row you want to erase is at the edge of the grid, then shift the character to push away the pixels.

Other special commands: Press I to invert the character, reversing all the pixels within the grid. R rotates the characters 90 degrees (turns the pattern on its side). Press R twice to turn a character upside-down. M gives you a left-to-right mirror image of the character, as if you picked up the character,

flipped it on its back, and put it back down.

Press C to copy the character you're editing to a new position in the character set. The current character replaces the character you select with the joystick or keyboard. Be careful with this, since you can unintentionally erase a cherished character. After the copy, the character you copied to is selected for editing.

Saving And Loading

When you're ready to save your character set, just press S, then type in a legal Atari filename at the prompt:

Save (Device:Filename)=

Include the D: for disk or C: for cassette. You can't save to the E: (screen editor) or S: (screen) devices, so if you forget the drive specification and your filename begins with E or S, FontMaker cancels the save. If you selected the save command by mistake and want to cancel it, just press RETURN when asked for the filename.

To load a character set into FontMaker, press L and enter the filename you used to save the character set. This will replace the character set in memory, so be sure to save the one you're working on if you want to keep it. Again, if you select this command by mistake, press RETURN at the prompt to cancel the load.

FontMaker is compatible with character sets created with "SuperFont" (originally published in COMPUTE!, January 1982, and also found in the *First Book of Atari Graphics*). FontMaker sets are also compatible with many other Atari character editors. We found that a FontMaker set can be loaded into the popular *Instedit* editor if you use a filename extension of .SET. Character sets created with the Iridis *Fontedit* also work with FontMaker.

If FontMaker can't save or load a character set due to a bad filename or problem with the disk, it displays the message I/O ERROR: PRESS RETURN. Press RETURN and try the save again, after you've figured out what went wrong.

When you're done editing characters, press the ESC (escape) key to exit FontMaker to DOS. Be sure you've saved the character set

you're working on if you want to keep it.

Using Fonts With BASIC

The Bytes option in FontMaker is primarily for programmers who want a list of the eight numbers that define a character, handy for changing just a single character in a program. But if you want to include an entire character set in your program, copying down these numbers can be tedious. Instead, you can use the two BASIC programs below, Programs 1 and 2, to add redefined character sets to BASIC.

Program 2, "Fontloader," is a subroutine you can merge with your BASIC program to load the character set into memory from disk. Just change the filename in line 1010 to the filename of your character set. The program loads the character set at the memory location CHSET, which is defined as eight pages back from the top of memory (location 106). Change this if you want to put your character set somewhere else. If a GRAPHICS command resets the character set, you can use POKE 756,CHSET/256 to re-point the character set vector to your RAM character set. Use POKE 756,224 to display the ROM character set, located at memory location 57344 (\$E000).

If you're using a *Translator* type of program on XL and XE computers, you can change CHSET (line 1030) to 57344 to load the character set directly into the RAM space corresponding to the position of the ROM character set. This keeps the character set in memory until you turn off the machine.

Use Program 3, "Character Set Datamaker," if you'd rather store your character set as part of your program in DATA statements. The Datamaker actually creates another program that you can merge with your own program. It creates DATA statements for only those characters that have been changed from the ROM image. Datamaker asks for the filename of your character set and a filename you'd like to use for the program it creates.

After using Datamaker, type NEW and use ENTER to load the program created by Datamaker. As with Program 2, you can change

CHSET in the program created by Datamaker if you want to determine yourself where the character set should go in memory. The program created by Datamaker copies the ROM set down to RAM (at CHSET), POKEs in the new characters, and switches the character pointer at 756 to the new set. Again, you can use POKE 756, CHSET/256 to reenact the set if your program somehow resets this pointer.

Next month we'll provide a program that lets you install an ANTIC 3 character set into your copy of *SpeedScript*, along with tips for creating readable characters and for using special characters for printer effects.

Program 1: FontMaker

Please refer to the "MLX" article in this issue before entering the following listing.

```
12280:167,125,032,076,054,032,252
12294:107,053,032,007,054,032,117
12300:062,059,032,100,059,032,070
12306:045,040,107,001,141,194,104
12312:040,032,242,040,032,132,027
12318:040,032,017,049,076,002,079
12324:051,167,044,168,040,032,040
12330:130,076,076,032,037,040,103
12336:169,012,133,002,169,137,208
12342:160,002,032,145,054,169,154
12348:002,133,002,076,076,111,042
12354:111,118,076,076,076,076,076
12360:114,032,070,121,032,067,024
12366:104,097,114,100,101,112,205
12372:032,066,114,097,110,110,101
12378:111,118,000,173,202,040,222
12384:201,005,200,018,173,194,117
12390:040,201,076,144,002,169,201
12396:000,076,169,001,076,032,244
12402:073,040,200,066,032,202,191
12408:054,032,202,054,169,120,247
12414:009,024,133,203,173,190,090
12420:040,009,002,133,204,162,170
12426:000,160,009,169,169,040,192
12432:145,203,000,152,041,002,120
12438:204,245,245,000,000,251
12444:240,032,073,040,200,000,015
12450:032,252,054,032,252,054,070
12456:076,000,000,000,000,000,000
12462:000,000,000,000,000,000,000
12468:000,000,000,000,000,000,100
12474:000,000,000,000,000,000,100
12480:000,000,000,000,000,000,100
12486:000,000,000,000,000,000,100
12492:000,000,000,000,000,000,100
12498:133,204,173,194,040,010,204
12504:030,204,010,030,204,010,200
12510:034,000,000,000,000,000,209
12516:195,040,133,203,165,204,152
12522:133,200,109,194,040,133,037
12528:204,076,032,200,040,160,200
12534:167,177,032,153,040,160,200
12540:133,177,040,134,014,045,003
12546:076,032,200,040,160,007,041
12552:105,169,040,145,202,134,124
12558:010,240,076,165,000,000,174
12564:105,200,133,203,165,000,175
12570:105,001,133,204,169,000,124
12576:141,193,040,162,004,173,241
12582:162,000,201,006,144,002,129
12588:162,000,133,000,133,193,202
12594:040,205,194,040,200,002,243
12600:009,120,145,203,230,193,204
12606:040,200,192,032,200,236,216
12612:010,160,200,105,000,000,174
12618:003,145,204,105,000,132,116
12624:204,202,200,218,173,202,007
12630:040,201,005,176,014,160,190
12636:000,105,119,049,201,010,144
12642:240,007,140,203,200,200,007
12648:244,160,000,209,173,194,051
```

12656:040,145,205,206,172,078,200
12660:044,245,956,052,104,101,090
12666:000,113,117,180,079,107,101
12672:000,079,114,111,119,110,160
12678:000,102,111,120,080,106,061
12684:117,109,112,101,194,000,107
12690:111,110,101,114,000,000,070
12696:000,000,000,000,000,000,152
12702:000,116,104,101,080,100,075
12708:000,122,121,000,100,111,203
12714:103,119,100,100,000,000,107
12720:000,000,000,000,000,000,179
12726:000,001,223,225,200,000,032,211
12732:224,000,000,105,079,107,035
12738:032,007,000,000,100,007,143
12744:114,007,000,111,114,072
12750:044,046,046,000,100,100,101
12756:160,049,032,130,024,032,157
12762:017,046,032,242,040,032,126
12768:113,040,175,135,007,248,104
12774:042,173,252,000,201,251,105
12780:200,043,174,120,002,109,204
12786:174,049,248,236,032,112,061
12792:053,142,051,200,024,100,047
12798:194,040,041,127,174,202,106
12804:040,224,000,144,002,041,213
12810:065,141,176,040,074,217,237
12816:049,173,132,002,248,251,095
12822:076,050,050,000,000,000,000
12828:041,127,201,004,170,013,170
12834:201,032,176,006,204,105,064
12840:004,076,047,000,056,233,054
12846:032,141,174,040,032,017,204
12852:049,032,242,040,113,054
12858:040,076,037,040,145,106,024
12864:056,233,016,133,204,141,079
12870:176,040,100,080,133,205,075
12876:141,90,040,241,044,040,075
12882:133,205,160,220,133,204,130
12888:160,000,177,205,145,203,110
12894:200,200,249,230,204,230,135
12900:200,165,206,201,229,200,135
12906:239,040,161,040,000,000,000
12912:000,167,003,141,020,200,152
12918:160,001,141,111,002,173,205
12924:176,040,050,233,004,141,034
12930:190,040,141,007,212,149,137
12936:200,200,161,040,000,000,000
12942:157,225,207,202,200,250,141
12948:173,190,040,133,204,173,053
12954:197,040,133,203,162,004,133
12960:160,000,132,100,000,205,252
12966:200,231,200,000,202,100,200
12972:246,024,169,120,133,200,057
12978:160,000,133,203,173,190,030
12984:040,000,003,133,204,133,203
12990:210,149,170,040,000,000,000
12996:024,165,207,145,203,073,245
13002:205,145,209,200,152,041,100
13008:000,000,000,165,207,073,102
13014:255,133,207,172,056,001
13020:232,169,050,141,000,200,000
13026:141,002,200,141,003,200,161
13032:167,003,141,009,200,141,135
13038:100,209,141,011,200,169,217
13044:014,141,192,002,169,090,004
13050:141,193,002,169,200,141,072
13056:194,002,169,030,141,195,227
13062:002,004,173,200,040,141,156
13068:000,200,173,200,000,141,156
13074:002,133,200,160,000,152,137
13080:170,133,203,145,203,200,094
13086:104,251,172,199,040,109,137
13092:047,051,145,203,200,232,146
13098:224,000,133,203,246,000,000
13104:144,144,246,000,000,004,064
13110:000,000,001,001,001,000,005
13116:255,255,255,000,000,000,057
13122:000,000,000,000,000,000,000
13128:001,255,000,000,000,000,000
13134:000,001,250,000,100,149,247
13140:054,141,200,040,169,024,110
13146:141,141,000,032,000,051,057
13152:174,120,100,000,000,000,000
13158:224,015,240,064,169,004,090
13164:133,207,141,031,200,105,225
13170:020,197,020,240,252,174,249
13176:201,000,141,051,020,172
13182:109,200,040,200,000,144,116
13188:007,201,000,176,093,141,233
13194:200,040,100,047,051,024,205
13200:109,90,040,201,024,144,101
13206:007,201,053,170,000,141,210
13212:199,040,032,000,051,190,100
13218:007,160,207,200,202,032,159
13224:112,033,173,202,002,201,193
13230:250,250,250,250,250,250,250
13236:200,170,050,173,199,040,010
13242:233,024,074,074,160,032,023
13248:093,040,200,014,192,006,243
13254:176,009,200,200,076,210,041
13260:051,152,050,233,006,160,102
13266:173,200,040,050,233,050,200
13272:074,074,176,232,050,169,223
13278:000,104,202,200,250,000,000
13284:160,040,153,100,040,032,070
13290:000,000,000,000,000,000,000
13296:201,040,201,010,200,000,150
13302:162,000,032,114,053,070,115
13308:053,051,100,000,141,000,194
13314:076,076,032,040,032,070,176
13320:200,100,000,000,000,000,000
13326:123,174,000,056,233,052,123
13332:174,050,052,221,000,032,123
13338:240,000,202,200,000,074,230
13344:076,076,032,040,032,070,176
13350:000,000,000,000,000,000,000
13356:100,050,052,072,100,000,205
13362:052,072,169,000,141,000,220
13368:000,000,000,000,000,000,000
13374:042,000,000,000,000,000,000
13380:027,000,120,005,124,007,000
13386:004,071,073,005,076,077,030
13392:002,076,067,067,066,070,034
13398:000,070,050,160,050,115,057
13404:055,194,054,244,050,040,244
13410:050,076,050,101,051,121,120
13416:050,200,049,142,050,173,010
13422:050,123,050,100,000,040,000
13428:055,105,053,100,050,000,000
13434:055,194,050,000,050,000,000
13440:050,103,050,240,050,105,142
13446:007,210,007,150,100,027,035
13452:007,007,007,007,007,007,007
13458:031,032,195,210,200,000,000
13464:050,027,000,027,020,027,079
13470:007,027,051,032,211,200,177
13476:200,190,100,050,032,030,205
13482:027,020,027,020,027,020,027
13488:155,211,200,195,195,197,050
13494:032,004,111,103,103,100,211
13500:101,032,032,032,032,195,100
13506:004,107,103,210,032,060,073
13512:100,007,110,107,105,200,000
13518:032,071,114,007,112,104,224
13524:105,009,110,032,077,111,239
13530:100,101,032,200,032,000,003
13536:105,009,107,105,105,000,000
13542:103,110,110,101,114,110,093
13548:102,032,032,032,032,032,244
13554:032,033,113,032,000,100,234
13560:100,101,110,100,100,100,000
13566:105,100,032,032,032,032,005
13572:032,032,207,200,210,201,120
13578:207,200,040,070,032,070,120
13584:105,100,032,032,032,032,074
13590:100,100,100,100,100,100,100
13596:007,114,101,032,032,032,102
13602:032,032,032,032,032,205,143
13608:032,077,100,114,114,110,001
13614:100,100,100,100,100,100,100
13620:112,121,032,032,032,032,157
13626:032,032,032,032,032,032,250
13632:194,032,046,121,114,101,102
13638:115,105,111,032,003,000,000
13644:110,100,032,102,111,110,100
13650:116,032,032,032,032,032,102
13656:004,032,074,111,007,100,194
13662:105,197,211,193,030,007,105
13668:000,100,100,100,100,100,100
13674:052,000,070,000,000,000,112
13680:162,000,160,000,150,200,100
13686:253,202,200,250,000,024,127
13692:165,000,165,000,133,000,000
13698:200,200,200,200,200,200,000
13704:133,204,133,210,162,002,212
13710:160,000,160,145,203,200,003
13716:200,251,236,204,202,200,170
13722:000,000,000,000,000,000,000
13728:002,040,173,203,040,201,010
13734:000,144,000,169,002,141,123
13740:002,040,032,107,050,032,214
13746:002,032,032,032,032,032,032
13752:017,049,076,173,040,002,057
13758:133,203,173,040,002,133,115
13764:204,160,003,169,144,145,047
13770:200,160,160,160,145,203,110
13776:200,173,202,040,145,203,110
13782:200,173,202,040,145,203,110
13788:200,173,202,040,145,203,110
13794:200,173,202,040,145,203,110
13800:200,173,202,040,145,203,110
13806:200,173,202,040,145,203,110
13812:200,173,202,040,145,203,110
13818:200,173,202,040,145,203,110
13824:200,173,202,040,145,203,110
13830:200,173,202,040,145,203,110
13836:200,173,202,040,145,203,110
13842:200,173,202,040,145,203,110
13848:200,173,202,040,145,203,110
13854:200,173,202,040,145,203,110
13860:200,173,202,040,145,203,110
13866:200,173,202,040,145,203,110
13872:200,173,202,040,145,203,110
13878:200,173,202,040,145,203,110
13884:200,173,202,040,145,203,110
13890:200,173,202,040,145,203,110
13896:200,173,202,040,145,203,110
13902:200,173,202,040,145,203,110
13908:200,173,202,040,145,203,110
13914:200,173,202,040,145,203,110
13920:200,173,202,040,145,203,110
13926:200,173,202,040,145,203,110
13932:200,173,202,040,145,203,110
13938:200,173,202,040,145,203,110
13944:200,173,202,040,145,203,110
13950:200,173,202,040,145,203,110
13956:200,173,202,040,145,203,110
13962:200,173,202,040,145,203,110
13968:200,173,202,040,145,203,110
13974:200,173,202,040,145,203,110
13980:200,173,202,040,145,203,110
13986:200,173,202,040,145,203,110
13992:200,173,202,040,145,203,110
13998:200,173,202,040,145,203,110
14004:200,173,202,040,145,203,110
14010:200,173,202,040,145,203,110
14016:200,173,202,040,145,203,110
14022:200,173,202,040,145,203,110
14028:200,173,202,040,145,203,110
14034:200,173,202,040,145,203,110
14040:200,173,202,040,145,203,110
14046:200,173,202,040,145,203,110
14052:200,173,202,040,145,203,110
14058:200,173,202,040,145,203,110
14064:200,173,202,040,145,203,110
14070:200,173,202,040,145,203,110
14076:200,173,202,040,145,203,110
14082:200,173,202,040,145,203,110
14088:200,173,202,040,145,203,110
14094:200,173,202,040,145,203,110
14100:200,173,202,040,145,203,110
14106:200,173,202,040,145,203,110
14112:200,173,202,040,145,203,110
14118:200,173,202,040,145,203,110
14124:200,173,202,040,145,203,110
14130:200,173,202,040,145,203,110
14136:200,173,202,040,145,203,110
14142:200,173,202,040,145,203,110
14148:200,173,202,040,145,203,110
14154:200,173,202,040,145,203,110
14160:200,173,202,040,145,203,110
14166:200,173,202,040,145,203,110
14172:200,173,202,040,145,203,110
14178:200,173,202,040,145,203,110
14184:200,173,202,040,145,203,110
14190:200,173,202,040,145,203,110
14196:200,173,202,040,145,203,110
14202:200,173,202,040,145,203,110
14208:200,173,202,040,145,203,110
14214:200,173,202,040,145,203,110
14220:200,173,202,040,145,203,110
14226:200,173,202,040,145,203,110
14232:200,173,202,040,145,203,110
14238:200,173,202,040,145,203,110
14244:200,173,202,040,145,203,110
14250:200,173,202,040,145,203,110
14256:200,173,202,040,145,203,110
14262:200,173,202,040,145,203,110
14268:200,173,202,040,145,203,110
14274:200,173,202,040,145,203,110
14280:200,173,202,040,145,203,110
14286:200,173,202,040,145,203,110
14292:200,173,202,040,145,203,110
14298:200,173,202,040,145,203,110
14304:200,173,202,040,145,203,110
14310:200,173,202,040,145,203,110
14316:200,173,202,040,145,203,110
14322:200,173,202,040,145,203,110
14328:200,173,202,040,145,203,110
14334:200,173,202,040,145,203,110
14340:200,173,202,040,145,203,110
14346:200,173,202,040,145,203,110
14352:200,173,202,040,145,203,110
14358:200,173,202,040,145,203,110
14364:200,173,202,040,145,203,110
14370:200,173,202,040,145,203,110
14376:200,173,202,040,145,203,110
14382:200,173,202,040,145,203,110
14388:200,173,202,040,145,203,110
14394:200,173,202,040,145,203,110
14400:200,173,202,040,145,203,110
14406:200,173,202,040,145,203,110
14412:200,173,202,040,145,203,110
14418:200,173,202,040,145,203,110
14424:200,173,202,040,145,203,110
14430:200,173,202,040,145,203,110
14436:200,173,202,040,145,203,110
14442:200,173,202,040,145,203,110
14448:200,173,202,040,145,203,110
14454:200,173,202,040,145,203,110
14460:200,173,202,040,145,203,110
14466:200,173,202,040,145,203,110
14472:200,173,202,040,145,203,110

```

14470:142,248,002,032,096,054,194
14480:167,000,135,120,005,140,231
14490:204,040,173,120,005,096,040
14500:003,097,110,101,032,040,191
14510:000,101,110,105,099,101,246
14520:098,099,105,100,101,110,212
14530:097,107,101,041,092,090,076
14540:169,160,140,054,032,130,123
14550:054,032,045,056,240,024,149
14560:201,044,240,028,201,003,242
14570:240,016,032,111,057,169,059
14580:000,141,200,040,140,011,022
14590:141,200,040,032,030,007,216
14600:076,037,040,076,111,097,153
14610:100,032,040,040,101,110,173
14620:105,099,101,050,070,105,092
14630:100,101,110,097,107,101,094
14640:041,042,000,149,223,160,131
14650:000,032,130,054,032,060,107
14660:050,240,024,201,009,240,062
14670:000,201,003,240,041,200,107
14680:111,057,169,004,141,200,107
14690:140,140,140,141,200,040,120
14700:032,030,057,076,037,040,040
14710:149,120,157,040,003,109,212
14720:000,157,040,003,123,204,130
14730:040,157,072,003,160,000,235
14740:157,073,003,149,003,157,099
14750:040,003,173,200,030,157,194
14760:247,003,169,000,157,070,026
14770:003,032,006,220,040,050,001
14780:173,195,040,157,060,003,204
14790:173,196,040,157,060,003,212
14800:169,006,157,072,003,196,142
14810:004,157,073,003,173,200,194
14820:040,157,060,003,032,006,203
14830:220,040,017,032,111,057,030
14840:040,032,076,123,030,030,020
14850:032,157,040,003,030,000,214
14860:220,040,032,111,057,169,040
14870:116,140,007,032,130,054,103
14880:032,029,050,076,030,040,211
14890:077,032,040,000,000,070,040
14900:002,050,253,032,000,000,219
14910:040,003,003,032,210,107,050
14920:212,213,210,200,000,067,040
14930:111,112,121,032,110,111,253
14940:032,119,104,105,000,100,210
14950:032,009,104,097,116,097,200
14960:099,110,116,060,000,161
14970:162,007,109,169,040,157,100
14980:100,040,202,016,247,169,030
14990:161,100,057,032,214,049,103
15000:162,007,109,105,040,157,104
15010:140,040,202,016,247,032,156
15020:167,056,032,119,054,226
15030:162,000,134,005,142,207,104
15040:000,109,169,040,032,250,196
15050:037,174,207,040,232,224,152
15060:000,200,239,032,029,050,046
15070:032,037,040,076,135,212,036
15080:169,000,133,213,032,170,201
15090:0217,032,230,216,100,000,009
15100:177,243,040,000,032,076,070
15110:054,200,200,246,041,127,120
15120:032,056,054,169,030,032,179
15130:056,054,056,173,000,190
15140:007,173,036,200,072,162,007
15150:000,076,224,002,225,002,075

```

Program 2: Fontloader

For instructions on entering this listing, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

```

J1000 REM FONTLOADER
J1010 OPEN "1,4,0,"D:SERIF
      SET REMOYER FILEN
      ARE HERE
J1020 X=1:REM FILENUM=16
J1025 DIM CID$(7):CDS="hh
      h":CID$(4)=CHR$(170)
      iCID$(5)="LV":iCID$(
      6)=CHR$(228)
J1030 CHSET=(PEEK(816)-8)*
      256:POKE 756,CHSET/2
      56:REM ADDRESS OF CH
      ARACTER SET. TRY 57
      344 ON XL'S WITH TRA
      NSLATOR
J1040 ICDB=834:ICBADR=836
      :ICBLEN=840
J1050 POKE ICBADR+X+1,CHSE
      T/256:POKE ICBADR+X,
      0

```

```

J1060 POKE ICBLEN+X+1,4:PD
      KE ICBLEN+X,0
J1070 POKE ICDBH+X,7:A=USR
      (ADR(CID$),X)
J1080 CLOSE #1
J1090 RETURN:REM REMOVE T
      HIS LINE TO USE THIS
      AS A STAND-ALONE PR
      OGRAM

```

Program 3: Character Set Datamaker

For instructions on entering this listing, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

```

J1100 OPEN #1,12,0,"E:"
J1102 GRAPHICS 1+16
J1105 DIM F$(14),DF$(14),T$(
      12),A(7)
J1110 POSITION 3,0:?"01"ch
      aracter set"
J1120 POSITION 5,2:?"02"IDE
      NTER:"?"06
J1130 ?"03:"THIS UTILITY CR
      EATES:"
J1140 ?"04:"A SET OF DATA S
      TATE:"
J1150 ?"05:"MENTS FROM A SA
      VED"
J1160 ?"06:"CHARACTER SET.
      IT"
J1170 ?"07:"OPTIMIZES 0Y ON
      LY"
J1180 ?"08:"LISTING CHARACT
      ERS"
J1190 ?"09:"NOT PRESENT IN
      THE"
J1200 ?"10:"STANDARD CHARACT
      ER"
J1210 ?"11:"SET."
J1220 ?"12:"?"06:"PRESS ANY
      KEY"
J1230 IF PEEK(53279)<>3 THE
      N 230
J1240 GRAPHICS 1+16
J1250 ?"13:"THE DATA STAN
      DENTS"
J1260 ?"14:"WILL BE WRITTEN
      TO"
J1270 ?"15:"DISK AS A list
      FILE"
J1280 ?"16:"USE enter TO ME
      ROE"
J1290 ?"17:"THE DATA WITH Y
      DUR"
J1300 ?"18:"PROGRAM."?"06:
      ?"19:"?"06:"PRESS ANY
      KEY"
J1305 POKE 82,0:POKE 87,0
J1310 ?"20:"CHR$(20):CHR$(15):
      "00":INPUT #1,T$:IF
      T$="" THEN 310
J1315 F$="D:"F$(3)=T$
J1320 ?"21:"CHR$(125):"?"06:
      ?"22:"(8 SPACES)"?"06:
      ?"23:"?"
J1330 ?"24:"CHR$(28):CHR$(15):
      "00":INPUT #1,T$:IF
      T$="" THEN 330
J1332 DF$="D:"DF$(3)=T$
J1335 ?"25:"CHR$(125):"?"06:
      ?"26:"(5 SPACES)"?"06:
      ?"27:"?"
J1340 INPUT SLINE
J1345 CLDSE #1
J1350 GRAPHICS 2+16:PDSITID
      N,5,6:?"06:"working
      (3 N):SETCOLD 4,3,4
J1370 OPEN #1,4,0,F$
J1380 TRAP 600:DOPEN #2,0,0

```

```

DF$;TRAP 40000
J381 ?"28:SLINE;"CHSET=(PE
      EK(100)-8)*256:FDR I=
      0 TO 1023:POKE CHSET+
      1,PEEK(57344+1):NEXT
      I"
J382 ?"29:SLINE+1;"RESTORE
      (SLINE+5
J383 ?"30:SLINE+2;"READ A:
      IF A=-1 THEN RETURN"
J384 ?"31:SLINE+3;"FDR J=0
      TO 7:READ 0:POKE CHS
      ET+A*8+J,0:NEXT J"
J385 ?"32:SLINE+4;"GOTO ";
      SLINE+2
J387 LINE=SLINE+4
J390 FDR I=0 TO 127:F=0
J400 FDR J=0 TO 7
J410 GET #1,A(J):A=
      12:0 IF A<>PEEK(57344+I*8
      +J) THEN F=1
J430 NEXT
J440 IF NOT F THEN 400
J445 LINE=LINE+1
J450 ?"33:SLINE;"DATA "I:?"
      02:I:"FDR J=0 TO 7:?"
      02:I:"A(J):NEXT J:"
      2"
J460 NEXT I:?"34:SLINE+1;"D
      ATA -I"
J470 POKE 82,2:GRAPHICS 0:
      ?"35:"All finished" Use
      ENTER "DF$
J480 ?"36:merge the file.
      "
J490 END
J500 POKE 82,2:GRAPHICS 0:
      ?"37:"ERRROR TRYING TO
      OPEN "DF$;"
J510 END

```

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 10954
Des Moines, IA 50340

or call the Toll Free number listed below.

Change Of Address. Please allow us 6-8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your COMPUTE! subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 months) US subscription to COMPUTE! is \$24.00 (2 years, \$45.00; 3 years, \$65.00). For subscription rates outside the US, see start page. Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of COMPUTE! or if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
1-800-247-5470
In IA 1-800-532-1272

Hi-Res Graphics Aid Routines

Jon Hylands

This handy utility makes it easy to perform sophisticated operations on Apple high-resolution graphics screens: inverting screens, copying screens, superimposing one screen on another, and more. It works on any Apple II-series computer with DOS 3.3 or ProDOS.

Like most personal computers, Apple II-series machines can display high-resolution color graphics. There are many commercial programs that let you draw, save, and reload hi-res screens. But few of them let you easily perform complex operations such as inverting an entire hi-res screen or superimposing one screen on another. "Hi-Res Graphics Aid" fills that gap. Though the program uses machine language for speed, you don't need to know ML to use it.

Type in and save the program below, then run it. The screen prompts are self-explanatory. Keep in mind that this is not a general-purpose drawing or design program; it performs large-scale tasks on existing graphics screens. Since the Apple can store two hi-res screens in memory at a time, most operations let you act on either screen 1 or screen 2.

When you run Graphics Aid, it displays a main menu of six selections. From this menu you can display a screen, edit a screen, load a screen, save a screen, display a disk catalog, or quit. The current selec-

tion is highlighted in inverse video. To choose a different selection, press the up-arrow or down-arrow keys (CTRL-K or CTRL-J on the Apple II+) and then press RETURN. Here's a brief description of the options:

Display screen. Enter 1 to display screen 1; 2 for screen 2.

Edit screen. This option displays a second menu with the following options:

- **Display screen.** Enter 1 or 2.
- **Invert screen.** Enter 1 or 2.
- **Copy screen.** Enter 1 to copy screen 1 to screen 2, or vice versa.

• **Superimpose screen.** Enter 1 to superimpose screen 1 on screen 2, or vice versa. Then choose the mode by pressing a number key from 1-3. Mode 1 is ORA mode; every pixel that's turned on in either screen remains on. Mode 2 is AND mode; only pixels that are on in both screens remain on. In Mode 3 (XOR), every pixel that's turned on in both screens will be turned off, and vice versa.

• **Color screen.** Choose screen 1 or 2, then enter a color number from 0-7.

• **Flip high bits.** Choose screen 1 or 2, then choose the mode by pressing a number key from 1-3. Mode 1 sets the high bits, mode 2 clears them, and mode 3 inverts them (on bits are turned off, and vice versa).

• **Swap screens.** Swap the contents of screen 1 and screen 2.

• **Return to command menu.**

Load to screen. Choose screen 1 or 2, then select drive 1 or 2 and enter the filename of the graphics file you wish to load.

Save screen. Choose screen 1 or 2, then select drive 1 or 2 and enter the filename you wish to use when saving the graphics screen to disk.

Catalog. Displays a disk catalog.

Quit. Exit to BASIC.

Hi-Res Graphics Aid

For instructions on entering this listing, please refer to "COMPUTER's Guide to Typing in Programs" in this issue of COMPUTE.

```
11 10 BA = 32768: FDR 1 = BA TO
BA + 212: READ ALCK = CK +
A: POKE 1,A: NEXT : REM L
DAD HR,CODE
20 10 IF CK < > 31397 THEN PRINT
"ERROR IN DATA STATEMENTS
": STOP
30 10 DATA 76,10,120,76,32,120,7
6,55
40 10 DATA 120,76,80,120,76,115,
120,76
50 10 DATA 151,120,166,255,173,0
0,192,173
60 10 DATA 82,192,173,87,192,189
,83,192
70 10 DATA 96,166,255,189,195,12
0,133,251
80 10 DATA 32,186,120,177,250,73
,255,145
90 10 DATA 250,32,177,120,200,24
5,96,145
100 10 DATA 255,189,195,120,133,
251,189,199
110 10 DATA 120,133,253,32,186,1
20,177,250
120 10 DATA 145,252,32,177,120,2
00,247,96
130 10 DATA 166,255,189,195,120,
133,251,189
140 10 DATA 190,120,133,253,166,
254,189,201
150 10 DATA 120,141,105,120,32,1
06,120,177
```



```

45 160 DATA 250,17,252,145,252,3
2,177,120
51 170 DATA 200,245,96,166,255,1
89,195,128
57 180 DATA 133,251,166,254,189,
205,128,141
63 190 DATA 141,128,189,209,128,
141,142,128
69 200 DATA 32,186,128,177,250,9
128,145
75 210 DATA 250,32,177,128,200,2
45,96,169
81 220 DATA 32,133,251,10,133,25
3,32,186
87 230 DATA 128,177,250,72,177,2
52,145,250
93 240 DATA 104,145,252,32,177,1
28,200,241
99 250 DATA 96,200,200,5,230,251
,230,253
105 260 DATA 202,96,160,0,132,250
,132,252
111 270 DATA 162,32,96,0,32,64,0,
64
117 280 DATA 32,0,17,49,81,0,9,41
2,290
123 290 DATA 73,0,120,127,120
13 300 TEXT : HOME : PRINT : PRI
NT CHR$(4):"PROM": PRINT
: REM INITIALIZATION
310 0% = CHR$(4):0% = CHR$(
17):0% = CHR$(27):L$ = "
-----":O = 1
1320 READ L: DIM X(L),T$(L,13
)
52 330 FOR J = 1 TO L: READ X(J)
: FDR I = 1 TO X(J): READ
T$(J,I): NEXT I
57 340 DATA 2,6,DISPLAY SCREEN,S
CREEN EDITOR,LDAO SCREEN,
SAVE SCREEN,CATALOG,QUIT
63 350 DATA 8,DISPLAY SCREEN,INV
ERT SCREEN,COPY SCREEN,SU
PERIMPOSE SCREEN,COLOR SC
REEN,FLIP HI BITS,SWAP SC
REENS,COMMAND MENU
10 360 READ X: DIM ERR$(X): FOR I
= 1 TO X: READ ERR$(I): N
EXT
13 370 DATA 13,,,WRITE PROTECTE
D,FILE NOT FOUND,VOLUME
MISMATCH,I/O ERROR,DISK F
ULL,FILE LOCKED,SYNTAX ER
ROR,,FILE TYPE MISMATCH
17 380 REM COMMAND MENU
21 390 HOME : TEXT :J = 1:IF$ = "
HI.RES COMMAND MENU": GOS
UB 640
51 400 IF I = X(J) THEN VTAB 10
+ X(J): END
56 410 ON 1 GOSUB 890,840,1290,1
360,1430
62 420 GOTO 390
67 430 REM GET A KEYSTROKE
72 440 A = 0: GET A$: IF A$ = $
THEN POP : RETURN
75 450 A = VAL (A$): RETURN
81 460 REM CENTER MESSAGE
87 470 VTAB V: HTAB ( INT ((40 -
LEN (M$)) / 2) + 1): PRI
NT M$: RETURN
93 480 REM DRAW A LINE
98 490 VTAB V: FOR I = 1 TO 4: P
RINT L$: NEXT I: RETURN
104 500 REM GET DRIVE
110 510 PRINT "DRIVE : ";D: CHR$(
13)
116 520 GOSUB 440: IF A$ = CHR$(
13) THEN A = 1
122 530 IF A < 1 OR A > 2 THEN 52
0
126 540 D = A: RETURN
132 550 REM GET PAGE
138 560 GOSUB 440: IF A < 0 OR A

```


The delay number must be expressed in hexadecimal (base 16). Thus, £01 pauses the printing for one second; £0F pauses for 15 seconds, and so on. By including delays in the character sequence, you can print information at any speed you like.

Pseudo-Keys

The £ character is an example of a Key Phantom pseudo-key. Instead of printing something on the screen, a pseudo-key performs a certain action. A second pseudo-key is the back-arrow key, located at the upper-left corner of the keyboard. When you include this character in a command sequence, Key Phantom waits for you to type a line from the keyboard. The input terminates when you press RETURN, just like INPUT in a BASIC program.

Since Key Phantom essentially types every character from the command file, you must keep in mind what would happen if you were typing those characters yourself. For instance, it's necessary to print a cursor down character before printing RETURN at the ends of the first three example lines. Otherwise you'd get a SYNTAX ERROR, since those lines don't contain BASIC commands. But no cursor down is used at the end of the last line: In this case you want to press RETURN at the end of the line to make the computer perform those actions.

Cursor control characters can be used for a variety of different effects. Just as in BASIC, you can move left, right, up, or down, go to the home position, clear the screen, and so on. The editor accepts any keys except DEL, CTRL, RUN/STOP, and RESTORE. Since control characters would garble the editor's screen display, it generally displays a reverse video < or > symbol to show that a control character was typed. If you need to perform an action not available from the keyboard, you can always execute a short PRINT statement. For instance, PRINT CHR\$(14);CHR\$(8) locks the keyboard into lower-case/uppercase mode.

Advanced Applications

Because the ML portion of Key Phantom is driven by the comput-

er's hardware interrupt routine, it can operate while a BASIC program is running. This means you can use it to feed input directly to a running BASIC program.

When would this be useful? To take a simple example, let's say you use a certain BASIC program frequently: It could be a checkbook program, an events calendar, or whatever. The program may begin by asking you to choose from several different options, input various items of information, and so on. By writing an appropriate command file for Key Phantom, you could make the computer load and run the BASIC program, select the option or options you want, and input as many items of information as needed. If you need to input new information at any point, the back arrow pseudo-key lets you do so. And when automatic control is no longer needed, the command file can terminate, leaving you in the BASIC program as usual.

If you're the type who likes to have several programming aids active at once, why not write a Key Phantom command file that automatically loads and activates all your favorite utilities at once? When you begin using Key Phantom, you'll probably think of many more uses as well.

Of course, since it uses memory from locations 49152-53247, this program is not compatible with utilities that use the same memory area. And you must be careful not to activate any other ML routines that disturb the 64's hardware interrupt vector at locations 788-789 (\$0314-\$0315). You should also look out for BASIC programs that begin by clearing the computer's keyboard buffer—to avoid losing information from the command file, you should pause Key Phantom for a few seconds (with the £ pseudo-key) when the program begins.

When feeding input to a program, you must keep in mind what sort of input the program expects. If the program accepts data with INPUT, you should terminate the corresponding data item with a carriage return. On the other hand, if the program accepts data with GET, you should not end the data with a carriage return. GET usually

takes a single keypress, not a keypress plus a carriage return.

Occasionally you may find a program that needs to use Key Phantom pseudo-keys for its own purposes. Then you'll have to change the pseudo-keys to some other character. This can be done by substituting different character codes in lines 570 and 590. The REMs in the program indicate which value belongs to which pseudo-key.

Commodore 64 Key Phantom

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing in Programs" published in this issue of COMPUTE!

```

JH 100 11=53368
HJ 110  CLOSE15:PRINT"[CLR]
      [2 DOWN]KEY PHANTOM
      [2 DOWN]"
QG 120  PRINT"[1] EDITOR[DOWN]":
      PRINT"[2] EXECUTE[DOWN]"
      PRINT"[3] EXIT[DOWN]"
CH 130  GOSUBB20:K=VAL(Q$):IFK<
      1ORK>3THEN130
AP 140  ON K GOTO150,460,510
PB 150  OPEN15,0,15,"I"
XQ 160  GOSUBB20:PRINT:PRINT$
FK 170  OPEN5,8,5,"8:COMMANDS,S
      W":GOSUBB20
HR 180  PRINT:PRINT$:IF AL<207
      THEN260
BK 190  CLOSES
QG 200  IF AL<63THEN110
JQ 210  PRINT"[DOWN]1] SCRATCH
      [DOWN]":PRINT"[2] APPEND
      [DOWN]":PRINT"[3] MENU
      [DOWN]"
RQ 220  GOSUBB20:K=VAL(Q$):IFK<
      1ORK>3THEN220
CF 230  ON K GOTO240,250,110
XD 240  PRINT"#15,"8:COMMANDS":G
      OTO160
MK 250  OPEN5,8,5,"8:COMMANDS,S
      A":GOSUBB20:GOTO100
QG 260  FOR LN=1TO6400:NEXT:LN=
      1
PG 270  PRINT"[CLR] [RIGHT]KEY P
      HANTOM "DG"[DOWN]":PRIN
      T"LINE - "LN"[DOWN]":LN
      =LN+1
HC 280  PRINT"[RVS] [OFF][LEFT]
      "
MP 290  GOSUBB20:K=ASC(Q$):IF K
      <32THENQ$="[RVS] [OFF]"
XQ 300  IF K<127AND K<168THENQ$
      ="[RVS] [OFF]"
HA 310  IF K=34THENQ$="[RVS]
      [OFF]"
CA 320  IF K=20THENPRINT"
      [2 LEFT]":K=LEN(L$)-1:
      L$=LEFT$(L$,K-(K<0)-2))
      GOTO280
RK 330  PRINTQ$
AQ 340  IF K<13THENL$=L$+CHR$(
      K):GOTO280
DF 350  PRINT:PRINT"[DOWN][RET]
      FOR NEXT LINE[DOWN]"
JC 360  PRINT"[DEL] TO REDO LIN
      E[DOWN]"
SR 370  PRINT"ANY OTHER TO EXIT
      [DOWN]"
GP 380  GOSUBB20

```


Screen Clock For IBM

Marc Sugiyama

Have you ever become submerged in a project while working on your computer and suddenly discovered it is hours past your bedtime? Or maybe you need to keep a detailed log of your worktime on the computer for business or tax purposes. If so, this utility is the answer—it constantly displays all this information and more on your monitor screen. It works with IBM PC and PCjr computers using DOS 2.0 or higher.

Large mainframe computers generally provide a *sysline* on the terminal screen which tells you the current date and time, who has logged on or off, and whether you've received any new electronic mail. Obviously, not all of these things apply to single-user personal computers, but some of the features would be nice to have.

"Screen Clock" is a short machine language program that prints the day of the week, date, current time, and log-on time at the top of the screen. This information appears no matter what else your computer is doing. You can be running a word processor, copying files, programming, or whatever—the day, date, and time will always be visible.

You might be wondering how it's possible to keep Screen Clock active while running another program; an IBM PC with PC-DOS isn't capable of multitasking. Screen Clock gets around this restriction by not using any PC-DOS function calls, relying instead on the BIOS (Basic Input/Output System) to handle the screen. This has several fortunate consequences:

- Sysline updates are not redirected to a file if you're using DOS file redirection.

- Sysline updates are not printed if you're echoing output to the printer. (But the sysline is printed if you press **PrtSc** for a screen dump.)

- Screen Clock always updates the current "active" screen. It doesn't matter if you switch from the monochrome monitor to the color monitor, change pages in the color screens, or even enter a graphics mode—the date and time are always there.

Winding Up The Clock

Type in the program listing below, save a copy on disk, then type **RUN**. The program is a BASIC loader that creates a machine language file on your disk with the filename **CLOCK.COM**. To start the clock, simply type **CLOCK** (upper- or lowercase is fine) at the **A>** DOS prompt. A sysline similar to this should appear on the top line of your screen:

Wed Jan 01, 1986 12:01A (00:37)

The day of the week, date, and current time are self-explanatory. The figure in parentheses is the elapsed time (in hours and minutes) since Screen Clock was started or reset. This "log-on" time runs up to 23 hours and 59 minutes, then rolls over to 00:00.

When you run Screen Clock from DOS, you can select various options by appending commands after typing **CLOCK**. Each command consists of a slash (/) symbol, a character, and sometimes a number. Here are the commands and options:

/C# (Chime) where *n* is an integer from 0 to 3. **/C0** means no chiming; **/C1** makes the clock chime hourly; **/C2** chimes every half-hour; and **/C3** chimes every 15 minutes. A chime is a low beep

which lasts for less than one second. Even if the screen updates are turned off, Screen Clock always chimes if you have told it to. The default is no chiming.

/Un (Update) where *n* is an integer from 1 to 9. This sets how often screen updates are to take place—*n* is the number of half-seconds between updates. The more frequent the updates, the more often the date and time are refreshed on the screen. However, more frequent updates also make other programs run more slowly. The default is equivalent to **/U2** (one second between updates).

/M (Military time). This selects military (24-hour) time.

/S (Standard time). This selects standard 12-hour time with an a.m./p.m. marker. Screen Clock defaults to standard time.

/R (Reset). This resets the log-on timer. Screen Clock automatically resets itself to 00:00 when first run.

For example, typing **CLOCK /U3/M/C1** at the DOS prompt loads and runs Screen Clock, sets updates every 1½ seconds, sets military time, and makes the clock chime every hour.

The Disappearing Clock

Occasionally, the Screen Clock sysline may get in the way. For example, it may hide text printed on the top line of the screen. You can make it disappear by pressing **CTRL** and both **SHIFT** keys simultaneously. Pressing this combination again turns the sysline back on.

Since Screen Clock maintains its own clock, it might not agree precisely with the DOS clock. Generally, it's never more than half a minute off.

Note that the day, date, and

time are reset every time you run Screen Clock. If you change the system date and time, you can reset Screen Clock by running it again. For example, the following would reset the display to 8:00 p.m. on February 14 (the A> prompts are supplied by DOS):

```
A> time 20:00:00
A> date 02-15-86
A> clock
```

The log-on time is not reset unless you append the /R command to CLOCK.

Although Screen Clock makes it appear that your computer is doing more than one thing at a time, it's important to remember that computers can really perform only one task at a time (a factor of the basic architecture of all personal computers to date). If the computer spends some of its time updating the sysline, that's time away from running the main program. Thus, the more often the sysline is updated, the more time it steals from the computer, and the slower the main program seems to run. However, the part of Screen Clock that takes the most time is printing the sysline on the screen. If screen updates are turned off, there is virtually no slowdown. So during heavy number crunching you might want to turn the sysline updates off.

I've been using Screen Clock quite a bit and haven't noticed much loss of performance at all. It seems that the computer spends a lot of its time waiting for input (from the keyboard, the disk drives, and so on); all we're doing is giving it something else to do in its "spare time." I have yet to find a program which doesn't work with Screen Clock.

As the power of personal computers increases, it becomes possible to include features once found only on large mainframe computers. A sysline such as Screen Clock is another step in this direction.

How It Works

Mainframe syslines are generally on the bottom row of the screen. The Screen Clock sysline, however, must be on the top row because there's no way via PC-DOS to keep the bottom row from scrolling. The sysline would keep traveling up the screen every time the screen was

scrolled. By placing the sysline on the top row, it can be refreshed each time it scrolls off the top of the screen.

The program itself is broken into two sections, resident and non-resident. The resident portion updates the internal counters, sounds the chimes, and updates the screen display. It's driven by the user interrupt 1Ch and is executed about 18 times a second. The nonresident part sets the initial date and time and changes the program's options.

When you execute CLOCK.COM, the program first checks to see if the resident portion is already installed. This is important only when the program returns control to DOS. Then it sets the current date and time and checks for any optional parameters. After this, the program is ready to return to DOS. If the program was already installed, it simply returns to DOS and does nothing else. If it needs to be installed, it first deallocates the environment space, then returns to DOS with the "terminate but stay resident" call to store the resident portion of the program safely in memory.

For instructions on entering the listing, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

IBM Screen Clock

```

E 100 CLS:LOCATE 10,10:PRINT"W
      string file ..."
E 110 OPEN "clock.com" FOR DUTP
      AS #1
E 120 FOR I=1 TO 1310:READ BYT
      E:CKSUM=CKSUM+BYT:IF BYT
      E<0 THEN FOR J=1 TO ABS(BY
      TE):PRINT#1,CHR$(0):NEXT
      J:GOTO 140
E 130 PRINT#1,CHR$(BYT)
E 140 NEXT I:CLOSE 1
E 150 IF CKSUM <> 124185 THEN P
      RINT"*** Error in DATA sta
      tement ***:KILL "clock.c
      om":STOP
E 160 PRINT:PRINT"File for clac
      k.com has been created."
      :END
E 200 OATA 233,51,4,74,97,110,
      32,78,181,78,32,77
E 210 OATA 97,114,32,65,112,11
      4,32,77,77,121,32,74
E 220 OATA 117,118,32,74,117,1
      89,32,65,117,183,32,83
E 230 OATA 101,112,32,79,99,11
      6,32,78,111,118,32,68
E 240 OATA 101,99,32,31,29,31,
      30,31,30,31,31,30
E 250 OATA 31,30,31,83,117,110
      ,32,77,111,110,32,84
E 260 OATA 117,180,32,87,101,1
      00,32,84,104,117,32,70
E 270 OATA 114,105,32,83,97,11
      4,32,-6,1,0,1,00
E 280 DATA 19,2,-5,240,18,0,1,
      -86,13,255,80,97
E 290 DATA 117,180,80,83,81,82
      ,86,87,85,30,6,140
E 300 DATA 200,142,216,142,192
      ,232,198,1,232,45,0,232
E 310 DATA 133,0,160,100,1,50,
      6,109,1,114,23,187
E 320 DATA 91,1,232,158,1,137,
      14,95,1,232,144,8
E 330 DATA 128,62,111,1,0,116,
      3,232,250,0,7,31
E 340 DATA 93,95,94,98,91,0
      ,8,207,180,2,205,22
E 350 DATA 36,7,60,7,116,6,190
      ,6,115,1,0,195
E 360 DATA 128,62,115,1,0,117,
      67,120,54,111,1,1
E 370 DATA 198,6,115,1,1,120,6
      2,111,1,0,116,4
E 380 DATA 232,197,0,195,108,1
      ,5,205,16,156,62,114,1
E 390 DATA 100,3,205,16,137,22
      ,112,1,100,2,106,-2
E 400 DATA 205,16,105,31,0,176
      ,32,180,14,205,16,226
E 410 DATA 250,180,2,138,62,11
      4,1,139,22,112,1,205
E 420 DATA 16,195,128,62,199,1
      ,255,116,25,160,199,1
E 430 DATA 58,6,198,1,119,5,25
      4,6,199,1,195,198
E 440 DATA 6,199,1,255,220,97,
      36,252,230,97,195,130
E 450 DATA 22,197,1,120,250,0,
      117,1,195,128,62,95
E 460 DATA 1,0,117,0,128,62,19
      6,1,0,116,62,195
E 470 DATA 108,250,1,116,50,12
      8,6,52,95,1,50,17,0
E 480 DATA 128,62,195,1,0,116,
      42,195,120,250,2,116
E 490 OATA 30,128,62,95,1,15,1
      17,0,128,62,196,1
E 500 DATA 0,116,22,195,128,62
      ,95,1,45,117,0,128
E 510 DATA 62,196,1,0,116,7,19
      5,190,6,196,1,0
E 520 DATA 195,198,6,196,1,1,1
      98,6,199,1,0,176
E 530 DATA 182,230,67,184,182,
      10,230,66,130,176,230,66
E 540 DATA 220,97,12,3,230,97,
      195,198,6,108,1,0
E 550 OATA 191,116,1,252,139,5
      4,102,1,209,230,209,230
E 560 OATA 129,198,63,1,185,4,
      0,243,164,139,54,97
E 570 DATA 1,209,230,209,230,1
      29,198,250,0,185,4,0
E 580 OATA 243,164,140,99,1,23
      2,22,1,184,44,32,171
E 590 DATA 160,101,1,232,12,1,
      160,100,1,232,0,1
E 600 OATA 176,32,170,139,14,9
      5,1,138,30,110,1,232
E 610 OATA 1,1,184,32,40,171,1
      87,184,1,232,50,0
E 620 DATA 180,1,232,242,0,176
      ,41,170,180,15,205,16
E 630 OATA 136,62,114,1,100,3,
      205,16,137,22,112,1
E 640 OATA 180,2,186,-2,205,16
      ,190,116,1,139,207,43
E 650 OATA 206,172,180,14,205,
      16,226,249,100,2,130,62
E 660 DATA 114,1,139,22,112,1,
      205,16,195,139,0,7,2
E 670 DATA 139,7,107,69,4,247,
      243,179,60,246,243,138
E 680 OATA 232,138,204,195,187
      ,91,1,232,16,0,115,3

```

```

M 698 DATA 232,43,8,187,184,1,
232,5,8,254,6,188
N 700 DATA 1,195,255,7,117,3,2
55,71,2,131,127,2
O 710 DATA 24,114,17,129,63,17
6,8,114,11,199,7,-2
P 720 DATA 199,71,2,-2,249,195
,240,195,255,6,182,1
Q 730 DATA 131,62,182,1,6,118,
6,199,6,182,1,-2
R 740 DATA 254,6,99,1,139,22,9
7,1,232,49,8,58
S 750 DATA 22,99,1,115,42,198,
6,99,1,1,255,6
T 760 DATA 97,1,131,62,97,1,12
,118,26,199,6,97
U 770 DATA 1,1,8,254,6,188,1,1
28,62,188,1,99
V 780 DATA 118,9,198,6,188,1,8
254,6,181,1,195
W 790 DATA 138,218,50,255,138,
151,58,1,128,251,2,117
X 800 DATA 16,246,6,188,1,3,11
7,9,128,62,188,1
Y 810 DATA 8,116,2,255,194,195
,212,10,5,48,48,134
Z 820 DATA 196,171,195,182,32,
128,252,1,116,18,182,65
[ 830 DATA 128,253,12,114,5,18
2,88,128,237,12,18,237
\ 840 DATA 117,2,181,12,138,19
7,232,217,255,176,58,178
] 850 DATA 138,193,232,289,255
,128,254,32,116,3,138,198
^ 860 DATA 178,195,82,181,113,
117,185,114,181,115,32,60
_ 870 DATA 79,83,32,58,46,48,3
2,111,114,32,97,98
` 880 DATA 111,118,181,46,13,1
8,36,78,111,119,32,185
a 890 DATA 118,115,116,97,188,
188,185,118,183,32,114,18
1
b 900 DATA 115,185,188,181,114,116,1
85
c 910 DATA 111,118,32,111,182,
32,67,76,79,67,75,46
d 920 DATA 13,18,36,39,32,117,
118,187,118,111,119,118
e 930 DATA 32,112,97,114,97,18
9,181,116,181,114,46,13
f 940 DATA 18,36,83,112,181,99
,185,182,121,32,97,32
g 950 DATA 118,117,189,98,181,
114,32,182,114,111,189,32
h 960 DATA 49,45,57,32,182,111
,114,32,39,85,39,32
i 970 DATA 115,119,185,116,99,
184,13,18,36,83,112,181
j 980 DATA 99,189,182,121,32,9
7,32,118,117,189,98,181
k 990 DATA 114,32,182,114,111,
189,32,48,48,51,32,182
l 1000 DATA 111,114,32,39,67,3
9,32,115,119,185,116,99
m 1010 DATA 184,13,18,28,-2,47
,188,48,285,33,68,8
n 1020 DATA 117,9,186,125,4,18
8,9,285,33,285,32,184
o 1030 DATA 8,55,285,33,136,22
,53,5,187,125,4,177
p 1040 DATA 4,211,235,67,137,3
8,51,5,184,28,53,285
q 1050 DATA 33,198,288,1,141,1
27,252,185,4,8,252,243
r 1060 DATA 166,131,249,8,116,
41,188,9,186,154,4,285
s 1070 DATA 33,184,28,37,186,2
84,1,285,33,38,7,232
t 1080 DATA 32,8,232,87,8,161,
44,8,142,192,188,73
u 1090 DATA 285,33,184,8,49,13
9,22,51,5,285,33,232
v 1100 DATA 8,8,232,63,8,184,8
,76,285,33,6,31
w 1110 DATA 188,8,285,26,137,2
2,91,1,137,14,93,1
x 1120 DATA 188,42,285,33,58,2
28,163,182,1,138,198,163
y 1130 DATA 97,1,136,22,99,1,1
98,6,181,1,19,129
z 1140 DATA 233,188,7,128,249,
99,118,7,128,233,188,254
[ 1150 DATA 6,181,1,136,14,188
,1,195,38,14,31,198
\ 1160 DATA 129,8,252,172,68,3
2,116,251,68,13,116,51
] 1170 DATA 58,6,53,5,116,241,
138,224,56,225,68,82
^ 1180 DATA 116,39,68,77,116,5
1,68,83,116,58,68,85
_ 1190 DATA 116,59,68,67,116,9
8,88,178,39,188,2,285
` 1200 DATA 33,88,138,212,188,
2,285,33,186,198,4,188
a 1210 DATA 9,285,33,31,195,38
,197,6,184,1,-2,38
b 1220 DATA 197,6,186,1,-2,235
,178,38,198,6,118,1
c 1230 DATA 1,235,178,38,198,6
,118,1,8,235,162,172
d 1240 DATA 68,49,114,21,68,57
,119,17,44,48,177,3
e 1250 DATA 138,224,218,228,2,
224,38,136,38,189,1,235
f 1260 DATA 136,186,221,4,188,
9,285,33,235,187,172,68
g 1270 DATA 48,114,13,68,51,11
9,9,44,48,38,162,197
h 1280 DATA 1,233,189,255,186,
8,5,188,9,285,33,235
i 1290 DATA 168,8

```

CAPUTE!

SpeedCalc Fixes

There are two errors in the DOS 3.3 listing for Apple SpeedCalc in the February 1986 issue (Program 1, p. 95). Lines 0FE2 and 11F2 from the listing cannot be entered as shown because smudged characters were inadvertently changed when retouched. The lines should read as follows:

```

0FE2: CA 10 EB CA 9D 00 02 E8 6E
11F2: F5 24 85 1E 20 22 0B 60 AB

```

These changes are not necessary if you have the February 1986 Apple COMPUTE! disk; the program on disk is correct.

ProDOS users will very likely encounter the message ERROR #56 when they attempt disk operations with that version. To correct this, you need to convert the binary (BIN) format file created by "Apple MLX" into a system (SYS) format file. To do this, first use the RENAME command to give the copy of SpeedCalc you entered with MLX the name SPEEDCALC.MLX. (Make sure that there is no file called just SPEEDCALC on the disk.) Then enter the three commands below, each on a separate line and each followed by pressing RETURN:

```

BLOAD SPEEDCALC.MLX
CREATE SPEEDCALC.SYS
BSAVE SPEEDCALC, $2000,$B3D67,
SYS

```

The new SPEEDCALC file now on the disk should function properly. Simply enter -SPEEDCALC to start it running. SpeedCalc already appears as a SYS file on the COMPUTE! disk for February, so this change is not necessary if you have the disk.

Speedy Strings For Commodore

The "Fast Disk Catalog" utility (Program 3, p. 66) from this article in the February issue does not work as listed. The SYS addresses assume that the machine language has been appended to the end of the program, as was done for Program 2. There are two possible solutions. You can change the lines below so that the machine language is POKED in and addressed properly:

```

GM 50 DIMFS(NM):A=0
CF 100 OPEN L,8,0,"80":SYS(AA):
CLOSE:L:CLOSE15
EC 120 PRINTX$:PRINT"[CYN]"*R

```

```

IGHT$(P$(C-1),2),C=B,C,
MM-C:B=C:AA=AA+16:GOTO80
XG 200 POKE987,78:POKE988,8:SY
S(AD)

```

Alternatively, you can append the machine language to the end of the program as was done for Program 2. This results in a shorter program that runs faster. To do this, don't change any lines in the program as listed. Instead, add the lines shown below, then type RUN 500. When the program ends, delete line 25 and all lines above 330, then immediately save a copy of the revised program.

```

GJ 500 POKE 45,(PEEK(45)+117)A
ND 255:POKE 46,(PEEK(46)
+1)-(PEEK(45)+117)
KC 510 POKE 47,(PEEK(45):POKE 4
8,(PEEK(46)):POKE 49,(PEEK
(45):POKE 50,(PEEK(46))
GJ 520 RESTORE:AD=PEEK(45)+256
*(PEEK(46)-373):FOR I=8 TO
0:367:READ D:POKE AD+I,
D:NEXT
GK 530 PRINT"[2 DOWN]DELETE LI
NE 25 AND ALL LINES ABOVE
330, THEN SAVE NEW V
ERSION":END
KC 1105 DATA 8

```



The World Inside the Computer

Fred Dignazio, Associate Editor

The Robot Inside You

Why are children so fascinated with robots? For that matter, why is everyone so fascinated with robots? The answer is that robots seem the most lifelike of all machines, and the most like real people.

When we see a little robot "toddler" like HEROjr sing songs to a trashcan, or when we watch a Movit robot like the WAO (pronounced "Wow") skitter crablike around the kitchen floor, avoiding tables and gargantuan human feet, we feel an uncanny thrill, as if we are watching a minor miracle. We know that these little machines are not alive. But they are sending visual cues of "aliveness" to the deepest parts of our brain. And these visual processing centers are flashing the message "Alive! Alive!" to the higher-level, rational center of our brain. We can deny that the machines are alive, but we will continue to feel that somehow they really are.

Young children most strongly and visibly reflect this sense of the aliveness of robots. Children's unfettered imaginations and their incomplete mastery of the scientific view of the world (so ingrained in us adults) cause them to see all sorts of objects as being alive—including teddy bears, dolls, shadows, imaginary friends, and, of course, robots. For them, the logic is simple: If it seems alive and acts alive, then it must be alive.

Not only do children ascribe the quality of aliveness to an object based on its behavior, but they also project a psychology—a personality or character—into the object. The object's personality stems partly from its behavior (if a robot sings to a trashcan, it must be a "silly" robot), but also as a projection of children's own personalities—their wishes, dreams, fears, and subconscious feelings. It would be impossible for children to articulate what these feelings and attributes are,

but they sense them immediately in a teddy bear, a beloved blanket, or an animated little robot.

The being that children see in these objects is very real, since it is a part of themselves. It may be their dark side, light side, or their happy or sad side, but it is an expression of a dimension of their own personality. Collectively these dimensions form children's complex, often contradictory humanity. In a real sense, then, there is a robot—a multitude of robots—inside every child; indeed, there are robots inside every one of us.

It is interesting to watch children struggle with the "Is it alive or not?" dilemma presented by today's robots and lifelike computer programs because we will all soon be facing this dilemma. In the coming years we adults will find our rational, scientific view of machines and other nonliving objects challenged by their increasingly lifelike characteristics. Their speech, mobility, sense of the world around them, and lifelike response are improving rapidly. All these traits will soon offer compelling evidence to our subconscious that the machines are really alive.

Kids feel this way already. For example, one little neighborhood boy of COMPUTE staffer Debi Nash played the new Activision game "Modern Computer People" in which little beings live inside the computer and interact with the world outside. The boy believed in the little creatures, and came by the Nash's house every day to talk with them and watch them live their lives. Unfortunately, one of the little people began to overeat. No matter what Debi and her family did, he kept stuffing himself. Suddenly the program crashed, and the person disappeared. Debi told me that telling the boy about the person's demise was as hard to do as

telling him that one of his friends had died.

Here in Birmingham, my six-year-old son Eric recently spent a couple days with A.G. Bear from Axlon Corp. A.G. talks in bear language but mimics human speech tones and rhythms with a little microchip in a voice box inside his chest. When Eric took A.G. to bed the first night, he had to take the voice box out of A.G. and leave it on the coffee table in the living room. Otherwise, A.G. would have begun talking every time Eric rolled over in bed or muttered something in his sleep. Eric happily took the bear to bed with him, but before he did he rushed over to the coffee table and wished the voice box good night. As I watched this little ritual from across the living room, I had the weird feeling that, for Eric, the voice box somehow held the little bear's electronic soul.

Last week, my nine-year-old daughter Catie and I were at the Bits & Bytes Computer Show for Children in Dallas, Texas. Together, she and I spoke to almost 400 schoolchildren about "Robot Pets & Friends." We demonstrated several popular robots, including Omnibot 2000, the Movit Family, and HEROjr, and we held a "Design Your Own Robot" contest which Catie judged.

The children's robot designs were original, diverse, and complex. To some extent, they resembled the robots that Catie and I had demonstrated and the robots of popular movies and TV shows. To a much greater extent, however, they were reflections of the children's own personalities. They were a revealing glimpse of the robots that dwell inside all of us. ©



Humanizing The User Interface, Part 2

Last month I wrote about several ways to make software easy to use. Now let's look at a model of human behavior that may hold the key for those who want to make computer programs that really stand apart from the crowd.

Psychologists and sociologists have spent a lot of time trying to figure out why people perform activities like rock climbing, playing chess, and performing other tasks for which the motivation appears to be internal (or *autotelic*). One behavioral scientist who has studied this area in some depth is Mihaly Csikszentmihalyi, whose book, "Beyond Boredom and Anxiety" (Jossey-Bass, Inc., 1975), makes some observations that are of great value to those of us involved with the design of user interfaces.

I believe that a good user interface makes the computer transparent to the user so he or she is free to interact purely with the application. Now look at some comments (typical of many) that Csikszentmihalyi collected from people involved with activities like playing chess: "The game is a struggle, and the concentration is like breathing—you never think of it. The roof could fall in and, if it missed you, you would be unaware of it."

A rock climber said: "You are so involved in what you are doing that you aren't thinking of yourself as separate from the immediate activity. You don't see yourself as separate from what you are doing."

The people described above are in "flow"—a state where action and awareness have merged. In this state the connection between the participant and the activity is so close that everything else seems to disappear.

Each of us has experienced flow at one time or another—perhaps while playing a game or watching a movie. But flow is a

hard state to maintain. For example, a computer user might experience flow with a video game, only to be bumped out of this state by the computer not responding fast enough, or by requiring the entry of a complex command that needs to be thought about consciously.

People who are not in a state of flow are often in a state of boredom or anxiety. Flow appears as a narrow band between the two.

When a person feels that the complexity of a task is too great for his or her skill level, a state of anxiety is produced. On the other hand, someone who has a lot of skill will be bored if the challenges are not great enough. Flow exists when the complexity of a task is appropriate for the skill of the participant.

As many readers will attest, good video games provide a nice model of the flow state. Most games allow the player to progress through a series of levels. The first level may induce anxiety in the novice player, but the player then acquires enough skill to enter a state of flow. The trick in good game design is to progressively increase the challenge level so that, as the player becomes more skilled, boredom doesn't set in.

This same model could be applied to the design of other types of programs. For example, a full-featured word processor might start out by encouraging the user to work with only a limited set of features, making the product easy to learn. As the user's skill increases more and more features can be revealed until the product is mastered.

This model of the flow state can help us understand another aspect of program design that seems to be misunderstood by some people. There is a difference between making a product easy to learn and making it easy to use. Ideally the

product should have both of these features. Instead we often encounter programs that provide a tremendous amount of hand-holding for the neophyte, but which are cumbersome to use once the product is mastered. Ease of learning, in this case, makes the product cumbersome to the proficient user.

Many of the complaints that have been lodged against the early releases of Macintosh software can be traced to this conflict. The use of pull-down menus is wonderful to the first-time user, since various options and commands can be presented in plain English. However, the physical act of moving the mouse to the menu bar, opening the menu, moving the mouse to the desired selection, and selecting this item, is cumbersome to the user who already knows what choice he or she wants to make. This is why an increasing number of Macintosh programs have single keystroke equivalent commands for menu selections. For example, a menu item named SAVE is a more obvious command than **Ctrl-S**, but the latter is an easier command to give, once the user knows its meaning.

Ideally, programs should provide a seamless transition from being easy to learn to being easy to use. Each computer user should be free to learn at his or her own rate.

In the past, such programs were hard to create, given the limitations of the computers on which they were running. Now that the Macintosh and the Amiga have entered the scene, computer horsepower is no longer an issue. The time has come for computers to become as easy to use as any other appliance we have at home. ☺



The Beginners Page

Tom R. Halfhill, Editor

More String-Slicing

Last month we saw how you can copy pieces of character strings using the `LEFT$` and `RIGHT$` functions found in versions of Microsoft BASIC. For even more flexibility, most Microsoft-style BASICs include a third function for extracting sections of strings. Called `MID$` ("mid-string"), this function lets you copy a section from the middle of a string.

The basic format is `MID$(string$,n1,n2)`, where `string$` is a string variable or literal string; `n1` is a number representing the beginning character position of the substring you want to extract; and `n2` is a number representing the number of characters in the substring you want to extract. For example:

```
10 A$="JAMES FENIMORE COOPER"
20 PRINT MID$(A$,7,8)
30 B$=MID$(A$,11,4)
40 PRINT B$
50 PRINT A$
```

When you run this program, the result is:

```
FENIMORE
MORE
JAMES FENIMORE COOPER
```

Line 20 prints the eight characters starting at position seven in `A$`, resulting in the substring `FENIMORE`. (Remember that spaces count as characters.) Lines 30 and 40 do much the same thing, but copy the four characters starting at position 11 into the string variable `B$` before printing them out. This method is useful if you need to print `B$` later in your program or manipulate `B$` in some other way. Line 50 shows that the `MID$` function, like `LEFT$` and `RIGHT$`, does not disturb the original contents of `A$`.

`MID$` is handy for so many different things that it's hard to come up with a generalized example. It can even be used to replace `LEFT$` and `RIGHT$`—for instance, `MID$(A$,1,10)` is equivalent to

`LEFT$(A$,10)`, and `MID$(A$,LEN(A$)-9,LEN(A$))` is the same as `RIGHT$(A$,10)`. One useful application of `MID$` is to store a bunch of short strings as a single long string, then pick out the substring you want with `MID$`. For example, let's say you're writing some sort of program that needs to print out the months of the year, perhaps as labels for a budget or chart. You could abbreviate the names of the months as equal-length substrings within one large string, like this:

```
10 M$="JANFEBMARAPRMAJUNJUL
AUGSEPOCTNOVDEC"
```

Now suppose that the numeric variable `M` contains the number of the month you need to print out—maybe as a result of an `INPUT` statement:

```
20 PRINT "WHICH MONTH TO
PRINT:
30 INPUT M
40 PRINT MID$(M$,M*3-2,3)
```

Depending on the user's response (1 for January, 2 for February, etc.), line 40 prints out the proper month name. Or you could print out all the months with a loop—`FOR M=1 TO 12:PRINT MID$(M$,M*3-2,3):NEXT M`.

Storing all the months in a single string and extracting the one you want with `MID$` is more efficient than using 12 separate strings for the same purpose. It's also more efficient in some ways than a string array (a subject we'll cover in a future column).

Atari And TI Strings

There are no `LEFT$`, `RIGHT$`, or `MID$` functions in TI BASIC or the Atari BASIC found on Atari 400/800, XL, and XE computers. These BASICs handle strings a little differently than Microsoft BASIC does. (Note that Microsoft BASIC is available on cartridge for Atari computers, and some BASICs available from independent suppliers

also support Microsoft-style strings.)

TI BASIC's statement for segmenting strings is `SEG$`. It works exactly like `MID$` in Microsoft BASIC—the statement `B$=SEG$(A$,11,4)` is equivalent to `B$=MID$(A$,11,4)`. You can simulate `LEFT$` with a statement in the form `SEG$(string$,n1,n2)`, where `string$` is the string you wish to manipulate, `n1` is the starting character position of the segment within the string, and `n2` is the number of characters you wish to print or copy. For example, the statement `B$=LEFT$(A$,6)` can be replaced with `B$=SEG$(A$,1,6)`.

Simulating `RIGHT$` is a bit more complicated. You need a statement in the form `SEG$(string$,LEN(string$)-n1,n2)`, where `n2` is the number of characters you wish to print or copy, and `n1` is `n2-1`. For example, `B$=RIGHT$(A$,6)` can be replaced with `B$=SEG$(A$,LEN(A$)-5,6)`.

Atari BASIC requires the same sort of manipulations. To print or copy any substring in Atari BASIC, simply specify the starting and ending character positions of the substring within the larger string. To translate `B$=LEFT$(A$,6)`, use `B$=A$(1,6)`. To simulate `RIGHT$`, use a statement in the form `string$(LEN(string$)-n,LEN(string$))`, where `string$` is the string you're manipulating and `n` is the number of characters you wish to print or copy minus one. For instance, to translate `B$=RIGHT$(A$,6)`, use `B$=A$(LEN(A$)-5,LEN(A$))`. To simulate `MID$`, use the statement `string$(n1,n2)`, where `n1` is the starting character position (just like `MID$`), and `n2` equals `n1` plus the number of characters you wish to print or copy minus one. Thus, the Microsoft statement `B$=MID$(A$,11,4)` is translated as `B$=A$(11,14)`. ☐



An April Trade Show Report

While most of the computer press converged on Lost Wages, Nevada in November of 1985 to attend yet another humdrum computer trade show, this columnist packed up his gear and headed for a counterculture communications fest held on the multileveled U.S.S. Flotsam, an ex-petro supertanker converted for use as a floating convention facility. Dubbed COMMDECKS 85 by show sponsor Aski Blok, it provided a fresh look at the lunatic fringe of computer communications.

There had been some doubt as to whether the show would get off the ground at all. Picket lines were set up by angry labor protesters who had been written bad redundancy checks by the show sponsor. The strikers were demanding even parity for all data transmitted to and from the show, making it almost impossible for exhibitors to set up their tables in the days preceding the show's opening. In the end, the demonstrators dropped their parity demands in exchange for 14 percent more than they had been receiving, plus additional time off in the form of one extra data bit and two stop bits.

It's hard to pinpoint the most memorable products of the show (since I spent most of my time recovering from the hors d'oeuvres and beverages served at evening press conferences), but I owe it to the readers of this column (and to the IRS) to take a shot at it.

Don't Just Ask For A Light

For the health-minded telecomputerist, Natural Language's line of optical wave modems are the first of the new "light" modems, transmitting 30 percent fewer characters than their wire-based counterparts. The new units are also said to aid the digestion of serial data (a.k.a. "number crunching") due to their high fiber optic content.

Setting a hard standard to beat

for intelligent modems is Thought System's new Kreskin 2400. How smart is it? The Kreskin reportedly can detect a busy signal before a call is actually made. Some recent prototypes also refuse to dial a remote Bulletin Board System if the unit's advanced circuitry senses there is nothing interesting to read or download on the BBS. The heart of the Kreskin is a superfast proprietary CPU chip capable of executing an infinite loop in 37 seconds. It translates the incoming stream of data into your choice of French, Italian, or Chinese (English is an extra-cost option).

No trade show would be complete without the obligatory raft of seminars and workshops, and COMMDECKS was no exception. Things did get off to a confusing start, however, when the kickoff session "The Future of VideoTex" turned out to be a panel discussion on merchandising VCRs and TVs in Dallas, Fort Worth, Austin, and Houston.

The highlight of the show's conference schedule had to be "Null Modems—Threat or Menace?"—a discussion of the trend toward violence in data communications hardware. The introduction of the Ninja, Terminator, and Rambo class of modem eliminators at the show further fueled the controversy to new heights.

Micro Telecomputing

With the price of mobile telephones dropping faster than a brick, Phylum Systems of Paramed, California figures to cash in big with a \$14.95 limited-distance mobile modem dubbed the Amoeba. Phylum's vice president of marketing, Ernest Flagella, says the single cellular unit will be shipping either "(1) real soon now, (2) in two weeks, (3) when the manual comes back from the printer, or (4) when Atari ST and Amiga owners stop

bickering over who bought the better machine."

Meanwhile, the Arapaho Indian Nation is entering the packet-switching network race to serve the communications needs of telecomputerists in the remote West. Bowing to the pressure of environmentalist groups, the Arapaho elders have agreed to house their telecommunications equipment inside structures disguised as totem poles. The job of cabling the poles has been awarded to RS-232 ace Louie "Bent Pin" Carson. Although Carson anticipates a high degree of difficulty in routing the cables within the highly confined spaces of the totems, he feels that a shot at everlasting fame is worth all of the headaches. When the job is done, Carson will have become the first man to wire a head for a reservation.

After-hours entertainment got physical on Friday night as anybody who is anybody attended a sports competition for manufacturers of multiuser LANs dubbed "Battle of the Network Stars." Over 50 teams vied for the coveted "Lord of the Rings" title, and the highly favored New York Subcarriers were disqualified in the early rounds for passing bad tokens in the relay.

To be perfectly honest, attendance at COMMDECKS 85 was far below the anticipated crowd of 25,000+. While hanging over the quarterdeck railing on the third day of the show, I bumped into promoter Aski Blok once more and quizzed him about the low number of attendees. "Well, it's really not too bad if you take everything into account," he said. "Our current location is kind of hard for people to get to. I think it would have been a lot more crowded if we hadn't cast off from the docks for the open C—the C programming workshop, that is."

©



Programming the TI

C. Regena

Creating Rhythms

A year ago, in the March 1985 issue of *COMPUTE!*, I published a program called "Drum Practice" for the TI-99/4A. That program was limited to quarter notes and quarter rests and the rhythms listed in DATA statements. This month, I'm offering a more complex program. You can create the rhythm for one measure by choosing notes and rests, and then the computer will play the rhythm for eight measures.

Lines 110-190 print the instructions. The different kinds of notes and rests available will appear at the bottom of the screen. You can use the arrow keys (on S and D) to move the red marker left or right to make your selection, then press the ENTER key. Your choice will then be printed on the staff above.

The available notes are a quarter note, two eighth notes together, one eighth note, two sixteenth notes, a dotted eighth note with a sixteenth note, a quarter rest, and an eighth rest.

Line 200 sets the time T equal to 75. If you want the rhythm to play faster decrease this number which represents the duration of a sixteenth note. Lines 210-240 read in from data (lines 260-330) the definitions for the graphic characters and define the characters from numbers 91 to 128. Line 340 defines R for a row number for the staff. Lines 350-360 define the red arrow used as a marker under the notes to be selected.

Lines 370-390 define variables in an array for the seven possible choices. Line 400 contains the data for this loop. For each of the choices from 1 to 7, D(C) is a value representing the counts—4 for one count, 2 for a half count. This variable is used to make sure the user makes a valid choice. For example, the computer will not allow a quarter note to be chosen if only a half

of a count is left in the measure. COL(C) is the column and is used to place the red marker.

The Rhythm Track

S\$(C) represents the durations when the rhythm is played. A sixteenth note factor is 1, so the quarter note is 4. Two eighth notes are 22, and one eighth note is 2. The two sixteenth notes are 11, and a dotted eighth with a sixteenth are 31. The rests are W and H. As the notes and rests are chosen, the string RHY\$ will add on values of S\$ (line 870).

Lines 1250-1390 play the rhythm. Line 1270 finds the length L of the string RHY\$. Line 1280 starts the loop for L number of times. Line 1290 looks at one character at a time of RHY\$. If the character A\$ is a letter, a rest is indicated so a frequency of 9999 with a volume of 30 is used. If A\$ is a number, that number is used as a factor times the previously defined T for the duration in the CALL SOUND statement, line 1310. If you prefer a different sound, change the frequency numbers in line 1310. I used the noise of -5 plus the frequency of 330. Line 1370 stops the sound so you can hear the different notes. The measure is played eight times.

Lines 420-440 wait for you to press ENTER before the program continues. Lines 460-500 clear the screen and print the notes using the redefined symbols. The lowercase y and z are typed by releasing the ALPHA LOCK key. Most of the symbols are typed by using the function key. Lines 510-630 draw the staff.

Lines 650-690 initialize variables for choosing the notes. COUNT and CHECK are used to determine how many notes and rests can be used in the measure. This measure is 4/4 time. A sixteenth note has a value of 1, so the

COUNT will go up to 16. CHECK is how many points are remaining in the measure. These numbers are used to verify which notes and rests can be used in the measure.

PLACE is the column number where the note or rest will start being drawn on the staff. The first note will start in column 8. PLACE is incremented depending on which note or set of notes is chosen. Lines 690-820 are the lines to get the user's choice. Line 730 makes sure the left arrow key (S), the right arrow key (D), or the ENTER key is chosen; all other keys are ignored by branching back to the CALL KEY statement. C is the choice number, and COL(C) is the column where the red marker appears for the choice.

It's Timing That Counts

Line 830 makes sure the choice is valid. The D timing value must be less than or equal to the number of sixteenth counts available. If the choice is not valid, the program plays an "uh-oh" sound and branches back to line 720, which is the CALL KEY statement to get another choice. Line 870 increments the RHY\$ string with the appropriate timing factors. Line 880 branches to the proper place for drawing the notes or rest and incrementing PLACE.

Lines 1220-1240 increment the COUNT and recalculate the CHECK time. If the measure is not full, the program goes back to get another choice. Lines 1250-1390 play the measure eight times. To stop the program, press FCTN BREAK.

If you have trouble running this program and get an error message in 220, 230, or 380, the actual cause of the error is most likely in the DATA statements of lines 260-330 or line 400.

All notes are placed at the E space of the staff, representing a



Two Checkers And A Manager

Anyone who spells as badly as I do is bound to love spelling checkers—and here are two new products that are getting a lot of attention.

Borland, the folks who brought you *Turbo Pascal* and the popular *SideKick*, have come up with another product headed for the best-seller's shelf. *Turbo Lightning* is a memory-resident spelling checker—it monitors every word you type and instantly beeps when you've made a mistake. Then, by pressing a key, you can call forth (in a box superimposed over your text) a list of the most likely correct words. It uses the 83,000-word Random House dictionary as its spelling authority. *Lightning* also has a thesaurus option which lets you select just the right word from a 50,000-word Random House thesaurus. All of this from within any program—word processor, spreadsheet, data management, or communications—just by pressing a few keys.

Here's how it works. *Lightning* stores a small dictionary in RAM. When installing the program, you must select one of three sizes: 6,000 words, 12,000 words, or 16,000 words. The larger the dictionary, the larger *Lightning's* vocabulary, and the less often it beeps for a word that is really correct. The trade-off, as always, is memory. As you type a word, *Lightning* consults the in-memory dictionary and beeps if there is no match. At this point, you may press the Alt-F9 keys to make the program consult the larger disk-based dictionary. *Lightning* then either confirms your spelling as correct or lists possible choices based on sound-alike words.

Two different disk-based dictionaries are available: one for hard disk systems and a smaller one for floppy disk computers. Since most of us have a small working vocabulary, the scheme of a RAM dictio-

nary supplemented by one on disk is quite workable.

A Flexible Engine

If you're thinking that a poor speller would be beeped to distraction, you are right. Fortunately, *Lightning* allows the auto-proof mode to be turned off; checking may then be requested on individual words or a screen at a time.

Borland plans to issue additional dictionaries and databases for use with the *Lightning* engine. In fact, any text-type data—even your own files—could be indexed and made accessible. *Turbo Lightning* is a sophisticated program with more potential than just a spelling checker. (*Turbo Lightning*, \$99.95, Borland International, 4585 Scotts Valley Drive, Scotts Valley, CA 95066.)

The second new spelling-checker is *Reference Set* from Reference Software. It too uses the Random House dictionary and thesaurus (what happened to Webster's?). *Reference Set* doesn't check each word as you type, but rather waits for you to request a spelling check by pressing Alt-D for dictionary or Alt-T for thesaurus. A window pops up over your text showing possible correct spellings (or alternate words); pressing a key deletes the old word and inserts the new one.

Although the dictionary is referenced from disk, the program maintains an index in memory so the time to locate a word, even with floppy disk, is typically less than a second. *Reference Set* includes two different sized dictionaries and thesauri for floppy and hard disk systems. The modest *Reference Set* engine—the memory-resident program that accomplishes the look-up—uses about 20K; by comparison, *Lightning* uses about 83K. (*Reference Set* Version 1, \$89.95, Reference Software, 2363

Boulevard Circle, Walnut Creek, CA 94595.) Both programs work best with a hard disk, but either may be used with a two-drive floppy system.

Automatic Stock Quotes

The "Manager" referred to in the title of this column alludes to a new program that works with the popular *Andrew Tobias' Managing Your Money* (see "IBM Personal Computing," December 1985). Called *Managing the Market*, it's a communications program that dials the Dow Jones News/Retrieval service, collects quotations, and updates the prices for the securities in an MYM portfolio. Pushing three or four keys dials the number, enters the password, selects the service, requests the quotes, updates the files, and disconnects. One nifty feature allows quotes to be ordered either by the percent change or by the absolute change—a real time-saver for those who monitor a lot of stocks. Output can be printed or saved for later analysis in a file readable by Lotus 1-2-3.

If you think this program would be too expensive to use with a modest portfolio, you may be pleasantly surprised. I've been updating about a dozen stocks, five days a week, after 6 p.m. when the rates are lowest, and the bill from Dow Jones is less than \$10 a month. *Managing the Market* comes with a temporary password and one hour of free time with Dow Jones, so you can begin using it right away. Of course, you must have a modem; the program supports all Hayes and Hayes-compatible modems as well as a dozen or so other makes. (*Managing the Market*, \$79.95, MECA, 12 Saugatuck Ave., Westport, CN, 06880.) ©



Binary Files, Unite!

I've had several people write me that various programs designed for use with binary (machine language) files don't work with Atari's *Macro Assembler (AMAC)*, *OSS's MAC/65*, or a couple of other assemblers. Or possibly a program will work with a small binary file produced by these assemblers, but not with a larger one. Why all these problems when the simple Atari Assembler/Editor cartridge works so well?

The root of the problem is the Atari Disk Operating System definition of a binary file, so let's examine that first. (Besides, maybe we'll learn a few extra goodies on the way.) A legal Atari binary file has the following format:

1. A header of two bytes, each with a value of 255 (hex \$FF).
2. Two more bytes indicating the starting address of a *segment* of the binary file. The two bytes are in standard 6502 low-byte/high-byte order.
3. Two more bytes indicating the ending address of that same file segment.
4. A sequence of bytes which constitute the actual binary code to be loaded into memory for the segment defined by the preceding four bytes. The number of bytes may be determined by subtracting the starting address from the ending address and then adding one.
5. If there are no further segments, there should be no more bytes in the file.
6. If there are more segments, then repeat this sequence of steps starting at either step 1 or step 2.

And that's it. A really neat, clean, format. Watch out for that last step, though. First, it says that the number of segments is theoretically unlimited. Second, it says that header bytes (dual hex \$FF bytes) may occur at the start of any segment. It also implies that there is no

particular order necessary to a binary file; it's perfectly OK to load the segment(s) at higher memory addresses before the one(s) at lower addresses.

RUN And INIT Vectors

Before moving on, there are two other niceties about DOS binary files worth knowing. When DOS loads a binary file (including an *AUTORUN.SYS* file at powerup), it monitors two locations. The simpler of the two is the *RUN* vector. Before DOS begins loading the binary file, it puts a known value into the two bytes at locations 736-737 (hex \$2E0-\$2E1). When the file is completely loaded (i.e., when DOS encounters the end of the file, step 5 above), if the contents of location 736 have been changed, then DOS assumes the new contents specify the address of the beginning of the program just loaded. DOS then calls the program (via a *JSR*) at that address.

The second monitored location is the *INIT* vector at address 738 (hex \$2E2). This vector works much the same as the *RUN* vector, but DOS initializes and checks it for each segment as the segments are loaded. If the *INIT* vector's contents are altered, then DOS assumes the user program wants to stop the loading process for a moment, long enough to call a subroutine. So DOS calls (via a *JSR*) at the requested address, expecting that the subroutine will return so the loading process can continue. This is a very handy feature. Most of you have probably seen it at work, such as when you run (or boot) a program which puts up an introductory screen (maybe just a title and a *PLEASE WAIT* message) and then continues to load.

The other important difference between the *RUN* and *INIT* vectors is that DOS leaves channel number one open while the *INIT* routine is

called. (DOS always opens and loads the binary file via this channel.) I suppose a really tricky program could close channel one, open up a different binary file, and then return to DOS. DOS would proceed to load the new file as if it were continuing the load of the original one. Most of the time, though, *INIT* routines should not touch channel one.

More On Segmented Files

Back to the main subject: Why do some programs have problems with binary files produced by some assemblers? Well, if all programs followed the complete binary file format as given by steps 1 through 6 above, there would probably be no incompatibilities. Unfortunately, many people who have used no assembler except the old cartridge have ignored segmented files. They have assumed that a binary file consists of steps 1 through 4, one time only, with a single large segment. Perhaps this is because many programmers first worked with Apple DOS, CP/M, and other operating systems with not-so-intelligent binary file formats. Or perhaps it is because the supposedly simple assembler cartridge is, in some ways, smarter than more advanced assemblers. In particular, the assembler cartridge will *not* produce multiple segments unless the programmer specifically asks for them (via an **=* directive to force a change to the location counter).

Yet other assemblers (including *AMAC* and *MAC/65*) never produce a segment longer than a particular size (usually a page—256 bytes—or less). If the programmer coded a longer segment, these assemblers automatically break it up into smaller pieces. Why? Probably to gain speed and lessen the work of assembly, since the assembler cartridge is doing a lot of work remembering the ending addresses

of segments.

Now, if my only concern were those few programs which don't properly load all binary files, I would simply have showed their authors the way to fix them. But there is a secondary advantage to programs which consist of larger segments: They load faster! Sometimes *much* faster. So this month I give you the BASIC program below, which takes any binary file and attempts to "unify" it. In particular, if the start address of one segment directly follows the end address of the preceding segment, they are consolidated into a single segment. And so on, so far as the space in BUF\$ allows.

And, last but not least, there's another minor bonus. Often, someone who writes an assembly language program purposely leaves space to be filled in later (e.g., by a filename, counter, etc.). If this reserved space occurs in the midst of code (probably not good practice, but it happens), it forces even the assembler cartridge to break the file into segments. But if the reserved space is significantly less than a sector (say under 30 bytes or so), it may be faster to let DOS load filler bytes. So you can change the value of the variable FILL in line 1160 (to 40, perhaps), and this program will automatically generate up to the specified number of fill bytes in an effort to better univf the file.

Whew! Was this month's topic too heavy for you? Then write me (P.O. Box 710352, San Jose, CA 95071-0352) with your suggestions.

for a topic. No treatises please. One or two pages works best. Thanks.

Binary File Unifler

For instructions on entering this listing, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

```

X 1110 REM allocate buffer
X 1120 REM
X 1130 BUFSIZE=FRE(0)-300
X 1140 DIM BUF$(BUFSIZE)
X 1150 DIM FILEOLD$(40),FILE
      NEW$(40)
X 1200 REM
X 1210 REM get file name
X 1220 REM
X 1230 PRINT "I need two fi
      le names: An existin
      g"
E 1240 PRINT " object file
      and a new file whic
      h"
E 1250 PRINT " will get th
      e 'unified' object c
      ode."
X 1260 PRINT
X 1270 PRINT "Existing file
      "
X 1280 INPUT #16,FILEOLD$
X 1290 PRINT "(5 SPACES)New
      file?" #1
X 1300 INPUT #16,FILENEW$
X 1400 REM
X 1410 REM open files, val
      date existing one
E 1420 REM
X 1430 OPEN #1,4,0,FILEOLD$
X 1440 GET #1,B$;FILEOLD$
      =B$;HIGH
X 1450 IF SEGLOW=255 AND SE
      GHIGH=255 THEN 1500
X 1460 PRINT :PRINT "Existi
      ng file: Invalid for
      mat"
X 1470 END
X 1480 REM input file okay
E 1490 REM
X 1500 OPEN #2,B$,0,FILENEW$
X 1510 PUT #2,SEGLOW:PUT #2
      ,SEGHIGH
E 1550 REM
X 1610 REM RM process a new or
      igin
X 1620 REM
X 1630 BUFFT=0

```

```

R 1440 BUF%=CHR$(3):BUF$(BUF
      SIZE)=CHR$(8)
H 1450 BUF%(2)=BUF%:REM zap
      buffer
K 1460 PUT #2,SEGLOW:PUT #2,
      $EGH1GH
D 1470 REM
M 1710 REM process a segment
      t
D 1720 REM
I 1730 GET #1,ENDLOW:GET #1,
      $ENDHIGH
H 1740 SEGSTART=SEGLOW+256*
      $EGH1GH:SEGENO=ENDLOW+
      256*$ENDHIGH
E 1750 SEGLEN=SEGENO-SEGSTAR
      RT+1
# 1760 REM read segment into
      a buffer
K 1770 FOR PTR=1 TO SEGLEN
R 1780 GET #1,BYTE:BUF$(BUF
      PTR+PTR)=CHR$(BYTE)
M 1790 NEXT PTR
R 1800 REM
R 1810 REM check head of ne
      xt segment
D 1820 REM
R 1830 GET #1,SEGLOW:GET #1,
      $EGH1GH
K 1840 IF SEGLOW+255 AND $E
      GH1GH=255 THEN GET #
      1,SEGLOW:GET #1,$EGH
      1GH
E 1850 SEGNEXT=SEGLOW+256*$E
      GH1GH
D 1860 GAP=SEGNEXT-SEGENO-1
K 1870 IF GAP<FILL OR GAP<0
      THEN 2000
R 1880 BUFPTR=BUFPTR+SEGLE
      N+GAP
I 1890 IF BUFPTR+256>BUFSIZE
      E THEN 2000
K 1900 GOTO 1700
E 2000 REM
M 2010 REM need to dump buf
      fer to
I 2020 REM prepare for new
      origin
K 2030 REM
E 2040 PUT #2,ENDLOW:PUT #2,
      $ENDHIGH
D 2050 FOR PTR=1 TO LEN(BUF
      %)
K 2060 PUT #2,ASC(BUF$(PTR)
      )
D 2070 NEXT PTR

```

**Write computer programs at home...
BE SUCCESSFUL WITH
COMPUTERS**
Self-paced, one-on-one instruction



Unique study program includes word-processing, bookkeeping, data base management, making lists, inventories and all about personal computers.

No experience needed!
Learn even before you
decide to a computer.

Professional, home and business applications.

HALIX INSTITUTE
CENTER FOR COMPUTER EDUCATION DEPT 81 4

YES Send me free information on how I can learn about computers and networking at home!

Name _____

Author's address:

Case

Discussion

DISK SALE!

Made by top USA makers, all brewed or brewed.
We use standards of major makers.
We don't just. The maker just
can't recognize them as among the **WORLD'S**
QUALITY PREMIUM WHISKY (most famous).
greatest 100% pure. **WORLD'S** best.
SATISFACTION GUARANTEED.

[illegible]

INCREDIBLE PRICES ON COMPUTER ACCESSORIES!

[illegible]

TOLL FREE (800) 343-0472
UNITECH IN MADE (817) 441-1000
200 HURLEY ST.

ATTENTION T.I. 99/4A OWNERS

- Diskettes - 59¢ each! Your choice SS or DD
- 512K Now Available for the 99/4A!
- 99/8 Level 4 Computer Upgrade Now Available
- Over 1500 Hardware and Software Accessories at Similar Savings.

**THE WORLD'S LARGEST
COMPUTER ASSISTANCE GROUP**

Now serving over 35,000 members worldwide with the best in technical assistance, service, and products for the Texas Instrument 99/4A Computer.

To become a member and receive newsletters, catalog, technical assistance and membership package, send \$10.00 for a ONE Year Membership to:

99/4A National Assistance Group
National Headquarters
P.O. Box 290812
Ft. Lauderdale, Florida 33329
Attention Membership Division
For Further Information Call 24 Hours
(305) 583-0467

Interactive Science Fiction Game For Commodore, Apple

PSI-5 Trading Company, from Accolade Software, is a science fiction "mini-drama," whose plot and outcome are contingent on the player's relationship with the story's characters. The game features detailed graphic depictions of 30 different characters who interact with the player through conversational text.

The story revolves around the PSI-5 Trading Company, a space freighter setting off on a mission to save the inhabitants of the Parvin Frontier from alien invaders. As captain of the ship, you must choose a crew of 5 from 30 applicants, each possessing special skills and a unique personality. The success of the mission hinges on the confidence you have in your crew to handle its responsibilities.

Suggested retail price for the Commodore 64 version is \$29.95. The Apple version retails for \$34.95.

Accolade Software, 20863 Stevens Creek Blvd., Cupertino, CA 95014.
Circle Reader Service Number 212.

Epyx Games Available For Amiga And Atari ST

Epyx has announced that two of its most popular computer games, and a microcomputer version of a classic mainframe adventure game, will be available for the Amiga and Atari ST this spring. In *Winter Games*, up to eight people can compete in seven events from the Winter Olympics. The original Commodore version of the game featured excellent graphics and sound. *Rogue* was originally a mainframe computer adventure game often played on college campuses. And the *Temple of Apshai* Trilogy offers a wide range of multiple dungeon levels, featuring 1400 separate chambers, plus enhanced high-resolution graphics.

Both versions of all three games are expected to retail for between \$19-\$39. Epyx, Inc., 1043 Kiel Ct., Sunnyvale, CA 94089.

Circle Reader Service Number 213.

Bantam Software Promotions

Bantam Electronic Publishing is offering software promotional deals for purchasers of *Sherlock Holmes In "Another Bow," The Fourth Protocol, and The Complete Scarsdale Medical Diet*. Through April 15, special rebate coupons can be used to take \$5 off the price of each of those programs. And, through March 31, Bantam will take entries in its Mystery Weekend contest, the winner of which will get a weekend for two in Boston to participate in a "mystery weekend" at the famous Parker House hotel. Special Holmes mystery pamphlets are available in many participating software stores. The pamphlets contain a mystery which you solve, and then submit to Bantam for a drawing in mid-April.

Bantam Electronic Publishing, Bantam Books, 666 Fifth Ave., New York, NY 10103.

Circle Reader Service Number 214.

Sports Tutorials Designed By The Pros

Avant-Garde has enlisted the help of three famous professional athletes in developing a line of sports tutorials. *Joe Theismann's Pro Football* offers advice on training and strategy to help develop quarterbacking techniques; improves overall football skills; and helps you understand the finer points of the game. Plays are illustrated through live-action diagrams. *Dave Winfield's Batter Up!* advises on pitchers, batting stance, swing height, grip and hitting strategy to help you develop expert batting techniques. The package also includes Winfield's book, *Batter Up! The Art of Hitting*, and a four-player batting game, *Sluggfest!* *Chris Evert-Lloyd Tennis* provides animated demonstrations of grip, stroke, game strategy, and specialized exercises. The program helps you learn the rules and choose the best equipment, and teaches concentration techniques to prepare you mentally for a match.

The Commodore 64 version of each program retails for \$34.95. The Apple II version (64K RAM minimum) and IBM-PC/PCjr version (128K RAM

minimum) retail for \$39.95.

Avant-Garde, 378 Commercial Blvd., Novato, CA 94947.

Circle Reader Service Number 215.

Infocom At The Big Top

In *Ballyhoo*, Infocom's new interactive mystery, you are a small-town circusgoer who sticks around after the show to explore the exotic back lot. What you discover is a mysterious underworld of crime and corruption, into which the circus owner's daughter has been kidnapped. In order to find her, you must solve a series of puzzles that are hidden among the circus folk.

Ballyhoo, one of Infocom's standard-level, all-text adventure games, is available for the Apple II-series and Macintosh; Atari XL/XE and ST series, Commodore 64/128 and Amiga, and the IBM PC and PCjr, for a list price of \$39.95.

Infocom, Inc., 125 Cambridge Park Dr., Cambridge, MA 02140.

Circle Reader Service Number 216.

New From Better Working

Word Processor with Spellchecker is the third product to be released in Spinnaker's Better Working line of home productivity software. The program is a full-function word processor, with a 50,000-word American Heritage Dictionary to catch spelling mistakes. It also features a 750-word personalized user dictionary, preview mode, microcommands for alternative print styles, and window-based menus and help screens.

The other titles in the Better Working series are *Spreadsheet* and *File and Report*. *Word Processor with Spellchecker* can perform mailmerge with *Better Working File and Report*. Each program is available for the Apple II series (\$59.95) and the Commodore 64/128 (\$49.95).

Better Working, Spinnaker, One Kendall Square, Cambridge, MA 02139.

Circle Reader Service Number 217.

Classified

SOFTWARE

TI-99/4A Software/Hardware bargains. Hard-to-find items. Huge selection. Fast service. Free catalog. D.I.C., Box 690, Hicksville, NY 11801

LOTTO PICKER. Improve your chances for those Million Dollar Jackpots! Picka LOTTO, WIN-4, & Daily Numbers. All USA & Can. games incl. Expandable IBM/C64/T99 \$29.95. Order Now! 1-800-341-1950 Ext. 77. Mail Orders: Ridge, 170 Broadway, #201-C, NYC, NY 10038 Catalog

PROJECT PLANNING/MANAGEMENT using the C64, SX, or C128. Data sheet for SASE. Prgm for \$106.95 (CA res. add 6% s/a tax). LANCO, Dept. C, Box 2059, Manteca, CA 95336

Genealogy Program for the C64. "FAMILY TREE" will produce Pedigree Charts, Family Group Records, Individual Files, Indexes, Searches of Ancestors. LDS version available. "The Best" genealogy program for the \$6, \$49.95. GENEALOGY SOFTWARE, POB 1151, PORT HURON, MI 48061, (519) 344-3990.

Animal Records maintained with "PETIKORE" for the C64. Produces Litter, Awards, Breeding Show, Individual Records, Pedigree Charts. \$69.95 GENEALOGY SOFTWARE, POB 1151, PORT HURON, MI 48061, (519) 344-3990.

FREE SOFTWARE CATALOG

Call Toll-Free 1-800-554-1162, Telex, Inc. Save 5% off retail prices. We carry SBI, Elect. Arts, Infocom, and many more!

COMMODORE: TRY BEFORE YOU BUY. Top 25 best-selling games, utilities, new releases. Visa/MasterCard. Free brochure. Rent-A-Disk, 908 9th Ave., Hantington, WV 25701 (304) 522-1666

DISCOUNT SOFTWARE. Amiga/Apple/Atari/C64-128/IBM PC-PCjr/TRS-80/Times/Sinclair. Free Catalog: WMF DATA SYSTEMS, 4 Butterfly Dr., Hauppauge, NY 11788

3 GAMES-PROBOS-Destroy a falling moon. 10101 - Play 21 against your computer. STALACTITE ATTACK-Try to survive in a cave with attacking stalactites-requires joystick. ALL THREE for 32K ATARI, ON DISK, \$5 check or M.O. to: M. A. Grossman, W205 N6844 Woodcut Dr., Menomonie Falls, WI 53051

Dr. T's MUSIC SOFTWARE

A music composition system with true word processing capabilities for APPLE and COMMODORE computers. Also algorithmic composition programs for computer generation of sequences. Fully editable Bach, Keyboard, Bass and Drum disks. Patch libraries available for Yamaha and Casio synthesizers. Reviewed in Jan '85. COMPUTE! (617) 244-6954, Dr. T's, Dept. C, 66 LOUISE RD., CHESHAM HILL, MA 02167

DATA BASE 130 FOR THE ATARI 130 XE

Data files up to 64K. Sort on any field. Print address labels or listings. \$25. Bill Van Sennet, P.O. Box 14585, Pittsburgh, PA 15234. Specify 810 or 1050

IBM PCjr OWNERS: We have the software you want. We also have complete 2nd drive add-ons or kits. FREE CATALOG. COWIL SOFTWARE, 1435 Burnley Sq. N., Columbus, OH 43229

FREE SOFTWARE for C64, C128, IBM & CPM. For info send large stamped (394) return envelope to PUBLIC DOMAIN USERS GROUP, P.O. Box 1642-A1, Orange Park, FL 32067

TI-99/4A QUALITY SOFTWARE for Business, Home and Entertainment ** BONUS Software Offer! ** Send for FREE catalog to MICRO - BIZ HAWAII, BOX 1106, Pearl City, HI 96782

TAX SPREADSHEET FOR C64, ATARI, TI: \$19.95 + \$2 s/t. Includes 1040, 2106, 2441, A.J.C.D.E.G.S.E.W.1040A. Yearly updates \$10. TI-99/4A w/ExtBasic, Atari w/48K or more. Specify disk or tape. Steve Kasek, 853 Diversity, St. Louis, MO 63126 (314) 561-2052

ATTENTION TI99/4A OWNERS!! See our ad in Product Mart. The 95/4A National Assistance Group (306) 583-6467

**** ATARI OWNERS ** QUEST FOR LIFE!** Realistic Adventure with weapons, killers and traps \$14.95 tape/disk. FREE CATALOG Octavian Software, Box 174, Republic, OH 44667

GAMES FOR C-64. ANY CHOICE \$19.95 T/D. Navy War, Star Trek, Football, Risk. Free game w/each buy. RK Games, 420 W. Central, Medicine Lodge, KS 67104 (316) 886-5097

MISCELLANEOUS

HELP IS ON THE WAY!

Just call 1-800-334-0868 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

Maxell MD1, \$1.29-MD2, \$1.99. Dysan 104/1D, \$1.79-104/2D, \$2.39. Shipping \$3.75. Also Verbatim, IBM, 3M, BASF. TAPE WORLD, 220 Spring St., Butler, PA 16001, 1-800-245-6908. Visa, MC.

EARN MONEY, PART OR FULL TIME, AT HOME with your computer-manual & forms-\$9.95. Write Computer Programs for Profit! How-to guide with forms, letters, tips-\$7.95. Also-Computer Consultant Handbook. How to earn \$ consulting with business-\$7.95. JV Tech, P.O. Box 563, Ludington, MI 49431

Discount computer printer ribbons for all makes/models. Etc. Epson 1500 Nylon \$6.99 Catalog. TWS 1314 4th Ave., Corvallis, PA 15108 (412) 262-1482 Visa or MasterCard

NEED THAT CIRCUIT DESIGNED into artwork, or that artwork fabbed into circuit boards, and the PCB's assembled? Then send requirements for free quote to the PC Master, 10221 Slater Ave., Suite 103-329, Fountain Valley, CA 92708

1985 INDEX TO ALL COMMODORE MAGAZINES. C64/128 disk with search/browse/print of articles, programs, reviews. Send \$16.95 ck or m.o. to: PCdex, Box 563, Dayton, OH 45409

IBM PCjr REPORT: THE NATIONAL NEWS-LETTER. PCjr-specific articles, reviews, Public Domain from across the nation. \$13./yr. PCjr CLUB, POB 95067, Schaumburg, IL 60193

RIIBBONS FOR MOST PRINTERS AT LOW PRICES! DELTA MICRONICS. C64 10933, IBM 16514-10933 (814) 455-3667

ROBOT KITS - THE MOVIE family of computer robots that you have been reading about. PEPPY - navigates by two-way sound sensor, tips around on three wheels, \$24.99 + \$2 s/t. AVOIDER - infra-red obstacle avoidance system, walks on six legs, \$44.99 + \$2 s/t. MR. BOOTSMAN - Wired remote control six leg robot. Two motor control, \$34.99 + \$2 s/t. Catalog \$6.00. Send orders to Computek, 530 South 6th St., Sunnyvale, WA 98944

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: \$25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add \$13 per line for boldface words, or \$55 for the entire ad set in boldface (any number of lines).

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.

Fare: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 15th of the third month preceding cover date (e.g., June issue closes March 15th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 9406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for omissions or errors of advertisements, but will attempt to screen out misleading or questionable copy.

MLX Machine Language Entry Program For Atari

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. You need to know nothing about machine language to use MLX—it was designed for everyone.

"MLX" is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for three numbers: the starting address, the ending address, and the run/init address. These numbers are given in the article accompanying the ML program presented in MLX format. You must also choose one of three options for saving the file: as a boot tape, as disk binary file, or as boot disk. The article with the ML program should specify which formats may be used.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a checksum number. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the DEL/BACK SPACE; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as 0123, the computer automatically prints the comma and goes on

to accept the next number. If you enter fewer than three digits, you can press the comma key, the space bar, or the RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

CTRL-S	Save
CTRL-L	Load
CTRL-N	New Address
CTRL-D	Display

To issue a command, hold down the CTRL key (CONTROL on the XL models) and press the indicated key. When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command (CTRL-S) to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember to make a note of what address you stop at. The next time you run MLX, answer all the prompts as you did before—regardless of where you stopped typing—then insert the disk or tape. When you get to the line number prompt, press CTRL-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press CTRL-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the MLX-format listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press CTRL-D, enter two addresses within the line number range of the listing. You can break out of the listing

display and return to the prompt by pressing any key.

Atari MLX: Machine Language Entry

For instructions on entering this listing, please refer to "COMPUTE's Guide to Typing in Programs" in this issue of COMPUTE.

```

N 100 GRAPHICS:0=L:PEEK13400
      +256*PEEK$5(1)+45*PO
      OL-1,71:POKE OL+2,6
N 110 POSITION B,0:?"MLX:"P
      OSITION 23,0:?"
      :PEEK 710,0:?"
N 120 ? "Starting Address":
      INPUT BEG:?" Ending
      Address":INPUT FIN:
      "Run/Init Address":IN
      PUT STARTADR
N 130 DIM A(6),BUFFERS(FIN-B
      EG+127),T(128),F(128),
      CIO$(7),SECTORS(128),O
      S$(56)
N 140 OPEN #1,4,0,"K":? :
      "Eape or Bsk":
N 150 BUFFER=CHR$(0):BUFFER
      =(FIN-BEG+30):BUFFERS=
      BUFFERS(2)=BUFFERS:SEC
      TORS=BUFFER%
N 160 ADDR=BEG:CIO$="hhh":CI
      O$(4)=CHR$(170):CIO$(5
      )="LV":CIO$(7)=CHR$(22
      B)
N 170 BE #1,MEDIA:IF MEDIA<
      30 AND MEDIA>60 THEN
      170
N 180 ? CHR$(MEDIA):? :IF ME
      DIA<30*(T+1) THEN BU
      FERS=##:GOTO 250
N 190 BEG=BEG+2:BUFFER=CHR
      $(0):BUFFERS(2)=CHR$(I
      NT((FIN-BEG+127)/128))
N 200 H=INT(BEG/256):L=BEG-H
      +256:BUFFER$(3)=CHR$(L
      ):BUFFER$(4)=CHR$(H)
N 210 PINT=BEG+L:IF PINT=FIN
      :L=L-1:IF L=0 THEN PINT=
      BUFFERS(15)=CHR$(L):BU
      FFERS$(6)=CHR$(H)
N 220 FOR I=7 TO 24:READ A:
      BUFFERS(I)=CHR$(A):NEXT
      I:DATA 24,96,169,60,1
      41,2,211,169,0,133,10
      ,169,0,133,11,76,0,0
N 230 H=INT(STARTADR/256):L=
      STARTADR-H+256:BUFFERS
      $(1)=CHR$(L):BUFFERS$(1
      9)=CHR$(H)
N 240 BUFFERS(23)=CHR$(L):BU
      FFERS(24)=CHR$(H)
N 250 IF MEDIA<30*(D+1) THEN
      N 360
N 260 ? :?"Boot Bsk or Bin
      ary Data":
N 270 BE #1,OTYPE:IF OTYPE<
      >60 AND OTYPE<70 THEN
      270
N 280 OTYPE=OTYPE:IF OTYPE
      =70 THEN 360
N 290 L=BEG+30:BUFFER=CHR
      $(0):BUFFERS(2)=CHR$(I

```

```

N1( (FIN-BEG+127)/128) )
H300 H=INT(BEG/256):L=BEG-H
#256:BUFFER(3)=CHR(L)
#310 1:BUFFER(4)=CHR(H)
INIT-STARTADR=H-INT(P
INIT/256):L=INIT-H#25
6:BUFFER(5)=CHR(L):B
UFFER(6)=CHR(H)
#320 RESTORE 330:FOR I=7 TO
30:READ A:BUFFER(I)=
CHR(A):NEXT I
#330 DATA 169,0,141,231,2,1
33,14,169,0,141,232,2,
133,15,169,0,133,10,16
9,0,133,11,24,96
#340 H=INT(BEG/256):L=BEG-H
#256:BUFFER(8)=CHR(L)
:BUFFER(9)=CHR(H)
#350 H=INT(STARTADR/256):L=
STARTADR-H#256:BUFFER
(22)=CHR(L):BUFFER(2
6)=CHR(H)
#360 GRAPHICS 0:POKE 712,10
:POKE 710,10:POKE 709,
2
#370 ? ADDR:"!":FOR J=1 TO
6
#380 GOSUB 570:IF N=-1 THEN
J=J-1:GOTO 380
#390 IF N=-1 THEN 720
#400 IF N=-12 THEN LET READ
=1:GOTO 720
#410 TRAP 410:IF N=-14 THEN
? :? "New Address":I
INPUT ADDR:1:GOTO 370
#420 TRAP 40000:IF N<-4 TH
EN 400
#430 TRAP 430: ? :? "Display
From":INPUT F1 ? :? "To
":INPUT T1:TRAP 32767
#440 IF FCBEG DR FFIN DR T
CBEG DR TFIN OR TCF T
HEN ? CHR(253):"At 1e
ast "BEG", Not More
Than " :FIN:GOTO 430
#450 FOR I=F TO T STEP 4: ?
? I:":FOR K=0 TO 3
:N=PEEK(ADR(BUFFER)+I
+K-BEG):TS="000":TS(4-
LEN(STR(N))):=STR(N)
#460 IF PEEK(724)<255 THEN
GET #1,A:PDP :PDP ? :
GOTO 370
#470 ? TS:":NEXT K: ? CHR
S(126):NEXT I: ? : ? B
D TD 370
#480 IF N<0 THEN ? :GOTO 37
0
#490 A(J)=N:NEXT J
#500 CKSUM=ADDR-INT(ADDR/25
6):256:FOR I=1 TO 6:CK
SUM=CKSUM+A(I):CKSUM=C
KSUM-256*(CKSUM/255):N
EXT I
#510 RF=128:GOSUB 0,200,12,
8:GOSUB 570:GOSUB 0,0,
0,0:RF=0: ? CHR(126)
#520 IF N<CKSUM THEN ? : ?
" Incorrect":CHR(253)
: ? :GOTO 370
#530 FOR I=15 TO 10 STEP -1:
GOSUB 0,10,10,W:EXT W
#540 FOR I=1 TO 6:POKE ADR(
BUFFER)+ADDR-BEG+I-1,
A(I):NEXT I
#550 ADDR=ADDR+A:IF ADDR<=F
IN THEN 370
#560 GOTO 710
#570 N=0:Z=0
#580 SET #1,A:IF A=155 OR A
=44 OR A=32 THEN 670
#590 IF A<32 THEN N=A:RETU
RN
#600 IF A<126 THEN 630
#610 GOSUB 670:IF I=1 AND T
=44 THEN N=-1: ? CHR(1
26):GOTO 690
#620 GOTO 570
#630 IF A<48 OR A>57 THEN 5
80
#640 ? CHR(A+RF):N=N#10+A
-48
#650 IF N>255 THEN ? CHR(2
53):A=126:GOTO 600
#660 Z=Z+1:IF Z<3 THEN 500
#670 IF Z=0 THEN ? CHR(253
):GOTO 570
#680 ? " ",:RETURN
#690 POKE 752,1:FOR I=1 TO
3: ? CHR(30):GET #1,T
:IF T<44 AND T<58 TH
EN ? CHR(A):NEXT I
#700 POKE 752,0: ? " :CHR(
126):RETURN
#710 GRAPHICS 0:POKE 710,26
:POKE 712,26:POKE 709,
2
#720 IF MEDIA=ASC("T") THEN
890
#730 REM 570
#740 IF READ THEN ? : ? "Loa
d File": ?
#750 IF DTYPE<>70 THEN 1040
#760 ? : ? "Enter AUTORUN.SY
S for automatic use"
: ? "Enter filename":I
INPUT TS
#770 F=TS:IF LEN(TS)>2 THE
N IF TS(1,2)<>"D:" THE
N F="D:" :F(3)=TS
#780 TRAP 870:CLOSE #2:OPEN
#2,B-4:READ,0,F: ? : ?
"Working..."
#790 IF READ THEN FOR I=1 T
O 6:GET #2,A:NEXT I:GO
TO 820
#800 PUT #2,255:PUT #2,255
#810 H=INT(BEG/256):L=BEG-H
#256:PUT #2,L:PUT #2,H
:H=INT(FIN/256):L=FIN-
H#256:PUT #2,L:PUT #2,
H
#820 GOSUB 770:IF PEEK(195)
>1 THEN 870
#830 IF STARTADR=0 OR READ
THEN 850
#840 PUT #2,224:PUT #2,2:PU
T #2,225:PUT #2,2:H=IN
T(STARTADR/256):L=STAR
TADR-H#256:PUT #2,L:PU
T #2,H
#850 TRAP 40000:CLOSE #2: ?
"Finished." :IF READ TH
EN ? : ? :LET READ=0:GO
TO 360
#860 END
#870 ? "Error "PEEK(195):"
trying to access": ? F
S:CLOSE #2: ? :GOTO 760
#880 REM 770
#890 IF READ THEN ? : ? "Rea
d Tape"
#900 ? : ? : ? "Insert, Rewin
d Tape." : ? : ? "Press PLAY
":IF NOT READ THEN
? : ? "RECORD"
#910 ? : ? "Press RECORD whe
n ready": ?
#920 TRAP 740:CLOSE #2:OPEN
#2,B-4:READ,12B,"C": ?
: ? "Working..."
#930 GOSUB 770:IF PEEK(195)
>1 THEN 960
#940 CLOSE #2:TRAP 40000: ?
"Finished." : ? : ? :IF R
EAD THEN LET READ=0:GO
TO 360
#950 END
#960 ? : ? "Error "PEEK(195
):" when reading/writi
ng boot tape": ? :CLOSE
#2:GOTO 890
#970 REM 570
#980 X=32:REM File #2, 320
#990 I=CBADR:B34:ICBADR=B36:I
CBLEN=B40:ICSTAT=B35
#1000 H=INT(ADR(BUFFER)/25
6):L=ADR(BUFFER)+H#2
56:POKE ICBADR+X,L:PO
KE ICBADR+X+1,H
#1010 L=FIN-BEG+1:H=INT(L/2
56):L=L-H#256:POKE IC
BLEN+X,L:POKE ICBLEN+
X+1,H
#1020 POKE ICCOM+X,11-4:REA
D A=USR(ADR(CIO),X)
#1030 POKE 195,PEEK(ICSTAT
):RETURN
#1040 REM 570
#1050 IF READ THEN 1100
#1060 ? : ? "Format Disk In
Drive 1 ? (Y/N)": ?
#1070 SET #1,A:IF A<>78 AND
A<>89 THEN 1070
#1080 ? CHR(A):IF A=78 THE
N ? 1100
#1090 ? : ? "Formatting..."
:X10 254,X2,0,0,"D": ?
"Format Complete": ?
#1100 NR=INT((FIN-BEG+127)/
128):BUFFER(FIN-BEG+
2)=CHR(0):IF READ TH
EN ? "Reading..." :GOT
D 1120
#1110 ? "Writing..."
#1120 FOR I=1 TO NR:S=1
#1130 IF READ THEN GOSUB 12
20:BUFFER(I*128-127)
=SECTOR:GOTO 1160
SECTOR=BUFFER(I*128
+127)
#1150 GOSUB 1220
#1160 IF PEEK(0STATS)<>1 TH
EN 1200
#1170 NEXT I
#1180 IF NOT READ THEN END
#1190 ? : ? :LET READ=0:GOTO
360
#1200 ? "Error on disk acce
ss." : ? "May need form
attng." :GOTO 1040
#1210 REM
#1220 REM 570
#1230 REM Drive ONE
#1240 REM Pass buffer in SE
CTORs
#1250 REM sector # in varia
ble S
#1260 REM READ=1 for read,
#1270 REM READ=0 for write
#1280 BASE=3#256
#1290 DUNIT=BASE+1:DCMDND=B
ASE+2:0STATS=BASE+3
#1300 DBUFLO=BASE+4:DBUFHI=
BASE+5
#1310 DBYTL=BASE+B:DBYTHI=
BASE+B+1
#1320 DAUX1=BASE+10:DAUX2=B
ASE+11
#1330 REM DIM DSKINVS(4)
#1340 DSKINVS="HLS":DSKINVS
(4)=CHR(228)
#1350 POKE DUNIT,I=ADR(5E
CTOR):H=INT(A/256):L
=A-256#H
#1360 POKE DBUFHI,H
#1370 POKE DBUFLO,L
#1380 POKE DCMDND,B-3:SEAD
POKE DAUX2,INT(B/256)
#1390 POKE DAUX1,5-PEEK(DA
UX2)*256
#1400 A=USR(ADR(DSKINVS))
#1410 RETURN

```

MLX Machine Language Entry Program

For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. You need to know nothing about machine language to use MLX—it was designed for everyone. At least 8K expansion memory is required.

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX for the 64 asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a checksum number. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the **INST/DEL** key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or **RETURN** key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581-584):

```

      U I O          7 8 9
H J K L become 0 4 5 6
M , .           1 2 3
    
```

64 MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

```

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display
    
```

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press **SHIFT-L** to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press **SHIFT-N** and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press **SHIFT-D**, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

64 MLX: Machine Language Entry

```

10 REM LINES CHANGED FROM MLX
   [SPACE]VERSION 2.00 ARE 550
   765,770 AND 860      REM 50
20 REM LINE CHANGED FROM MLX V
   ERSION 2.01 IS 300 REM 147
100 PRINT"[CLR]";CHR$(142);
   CHR$(8);POKE53281,1:POKE5
   3280,1      REM 67
    
```

```

101 POKE 768,52:REM DISABLE RU
   N/STOP      REM 119
110 PRINT"[RVS]";      REM 176
   REM 176
120 PRINT"[RVS]";[14 SPACES]
   [RIGHT][OFF]8*12[RVS]
   [RIGHT][RIGHT]2 SPACES]
   8*1[OFF]8*12[RVS]2[RVS]
   [14 SPACES]";      REM 250
130 PRINT"[RVS]";[14 SPACES]
   [RIGHT]80[RIGHT]
   [2 RIGHT][OFF]2[RVS]2
   8*1[OFF]8*12[RVS]2
   [14 SPACES]";      REM 35
140 PRINT"[RVS]";[41 SPACES]"
   REM 120
200 PRINT"[2 DOWN][PUR][BLK] M
   ACHINE LANGUAGE EDITOR VER
   SION 2.02[5 DOWN]";REM 238
210 PRINT"[53]";[2 UP]STARTING AD
   DRESS[8 SPACES]";[9 LEFT]";
   REM 143
215 INPUTS:F=1-F:CS=CHR$(31+11
   9*F)      REM 166
220 IF$256OR($4096AND$4915
   2)OR$532477HENGOSUB3000:G
   OTO210
225 PRINT:PRINT:PRINT REM 180
230 PRINT"[53]";[2 UP]ENDING ADDR
   ESS[8 SPACES]";[9 LEFT]";:I
   NPUTS:F=1-F:CS=CHR$(31+119
   *F)      REM 20
240 IF$256OR($4096AND$4915
   2)OR$532477HENGOSUB3000:G
   OTO230
250 IF$SYNHPRINTCS;"[RVS]END
   ING < START[2 SPACES]";:GOS
   UB1000:GOTO 230 REM 176
260 PRINT:PRINT:PRINT REM 179
300 PRINT"[CLR]";CHR$(14):AD$=
   REM 56
310 A=1:PRINTRIGHT$("0000"+MID
   $(STR$(AD),2,5)):"";      REM 33
315 FOR$=ATO6
320 GOSUB570:IFN=-1THENJ=J+1:G
   OTO320
390 IFN=-211THEN 710 REM 62
400 IFN=-204THEN 790 REM 64
410 IFN=-206THENPRINT:INPUT"
   [DOWN]ENTER NEW ADDRESS:";2
   REM 44
415 IFN=-206THENIF$3<SOR$23*H
   ENPRINT"[RVS]";[OUT OF RANGE"]
   :GOSUB1000:GOTO410 REM 225
417 IFN=-206THENAD=23:PRINT:GO
   TO310 REM 238
420 IF B<-196 THEN 400
430 PRINT:INPUT"DISPLAY:FROM";
   F:PRINT,"TO";INPUT"
   REM 234
440 IF$[SORF]EORT[SORF]ETHENP
   INT"AT LEAST";8;"[LEFT], N
   OT MORE THAN";I:GOTO430
   REM 159
450 FORI=PTOTSTEP6:PRINT:PRINT
   RIGHT$("0000"+MID$(STR$(I
   ),2,5)):"";      REM 30
451 FORW=0TOS:N=PEEK(1+K):PRIN
   TRIGHT$("00"+MID$(STR$(N),
   2,3)):"";      REM 66
    
```

Save Your Copies of COMPUTE!

Protect your back issues of *COMPUTE!* in durable binders or library cases. Each binder or case is custom-made in flag-blue binding with embossed white lettering. Each holds a year of *COMPUTE!*. Order several and keep your issues of *COMPUTE!* neatly organized for quick reference. (These binders make great gifts, too!)



Binders	Cases:
\$8.50 each;	\$6.95 each;
3 for \$24.75;	3 for \$20.00;
6 for \$48.00	6 for \$36.00

(Please add \$2.50 per unit for orders outside the U.S.)

Send in your prepaid order with the attached coupon

Mail to: Jesse Jones Industries
P.O. Box 5120
Dept. Code COTE
Philadelphia, PA 19141

Please send me _____ *COMPUTE!* cases ☐ binders.
Enclosed is my check or money order for \$ _____ (U.S. funds only.)

Name _____
Address _____
City _____
State _____ Zip _____

Satisfaction guaranteed or money refunded.
Please allow 4-6 weeks for delivery.

```

460 GETA$;IFA$="*";THENPRINT:PRI
NT:GOTO318      :rem 25
470 NEXTI:PRINTCHR$(28);:NEXTI
:PRINT:PRINT:GOTO318
      :rem 50
480 IFN<0 THEN PRINT:GOTO318
      :rem 168
490 A[J]=NEXIJ      :rem 199
500 CKSUM=AD-INT(AD/256)*256:F
ORI=1:TO6:CKSUM=(CKSUM+A(I)
)AND255:NEXT      :rem 280
510 PRINTCHR$(16);:GOSUB78:PR
INTCHR$(146);      :rem 94
511 IFN=-1:THENA=6:GOTO315
      :rem 254
515 PRINTCHR$(20);IFN=CKSUMTHE
N538      :rem 122
520 PRINT:PRINT"LINE ENTERED W
RONG : RE-ENTER":PRINT:GOS
UB1808:GOTO318      :rem 176
530 GOSUB2000      :rem 210
540 FORI=1:TO6:POKEAD+I-1,A(I):
NEXTI:POKE54272,0:POKE54273
,0      :rem 227
550 AD=AD+6:IF AD<E THEN 318
      :rem 212
560 GOTO 718      :rem 188
570 N=0:I=0      :rem 80
580 PRINT"£4$";      :rem 81
591 GETA$;IFA$="*";THEN581
      :rem 95
592 AV=(A$="M")-2*(A$=",")-3*
(CKSUM=AD)-4*(A$="J")-5*(A$=
"K")-6*(A$="L")      :rem 41
593 AV=AV-7*(A$="U")-8*(A$="I"
)-9*(A$="O"):IFA$="H":THENA
$="H"      :rem 134
594 IFAV<0:THENA$=CHR$(48+AV)
      :rem 134
595 PRINTCHR$(20);:A=ASC(A$):I
FA=1308A=44ORA=32:THEN670
      :rem 229
596 IFA=128:THENN=A:RETURN
      :rem 137
600 IFA<20 THEN 638      :rem 10
610 GOSUB690:IFI=1ANDI=44:THEN
N=1:PRINT"[OFF] [LEFT]
[LEFT]":GOTO690      :rem 62
620 GOTO570      :rem 109
630 IFA<48ORA>57:THEN580
      :rem 105
640 PRINTA$;:N=N*10+A-48
      :rem 106
650 IFN>255 THEN A=20:GOSUB180
:GOTO600      :rem 229
660 I=2+1:IF2<3:THEN580      :rem 71
670 IF2=0:THENGOSUB1800:GOTO570
      :rem 114
680 PRINT",";:RETURN      :rem 248
690 S$=PEEK(209)+256*PEEK(210)
+PEEK(211)      :rem 149
691 FORI=1:TO3:T=PEEK(S$+I)
      :rem 67
695 IPT<44ANDT<58:THENPOKES$-
2,32:NEXT      :rem 285
700 PRINTLEFTS(" [3 LEFT] ",1-1)
:RETURN      :rem 7
710 PRINT"[CLR] [RVS]*** SAVE *
** [3 DOWN]"      :rem 236
715 PRINT"[2 DOWN] [PRESS [RVS]
RETURN [OFF] ALONE TO CANCE
L SAVE] [DOWN]"      :rem 106
720 F$="":INPUT"[DOWN] FILENAM
E":F$=F$+"*";THENPRINT:PRI
NT:GOTO318      :rem 71
730 PRINT:PRINT"[2 DOWN] [RVS]T
[OFF] APE OR [RVS] [D] [OFF] ISR
: (T/D)"      :rem 228
740 GETA$;IFA$<"*":ANDA$<"D":C
HNR740      :rem 36
750 DV=1-7*(A$="D"):IFDV=8:THEN
      :rem 89
F$="0":F$=OPEN15,8,15,"*"
+F$+CLOSE15      :rem 212
760 TS=F$:ZK=PEEK(53)+256*PEEK
(54)-LEN(T$):POKE782,ZK/25
6      :rem 3
762 POKE781,ZK-PEEK(782)*256:P
OKE780,LEN(T$):SYS65469
      :rem 109
763 POKE780,1:POKE781,DV:POKE7
82,1:SYS65466      :rem 69
765 K=SP:POKE254,K/256:POKE253,
K-PEEK(254)*256:POKE780,25
3      :rem 17
766 K=E+1:POKE782,K/256:POKE78
1,K-PEEK(782)*256:SYS65496
      :rem 235
770 IF(PEEK(783)AND1)OR(19AND
ST)THEN780      :rem 111
775 PRINT"[DOWN] [DONE. [DOWN]:G
OTO318      :rem 113
780 PRINT"[DOWN] ERROR ON SAVE.
[2 SPACES] TRY AGAIN. *IFDV
=1:THEN720      :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E
2$:PRINTI$;E2$:CLOSE15:GO
TO720      :rem 103
790 PRINT"[CLR] [RVS]*** LOAD *
** [2 DOWN]"      :rem 212
795 PRINT"[2 DOWN] [PRESS [RVS]
RETURN [OFF] ALONE TO CANCE
L LOAD]"      :rem 82
800 F$="":INPUT"[2 DOWN] FILEN
AME":F$=F$+"*";THENPRINT:G
OTO318      :rem 144
810 PRINT:PRINT"[2 DOWN] [RVS]T
[OFF] APE OR [RVS] [D] [OFF] ISR
: (T/D)"      :rem 227
820 GETA$;IFA$<"*":ANDA$<"D":C
HNR820      :rem 34
830 DV=1-7*(A$="D"):IFDV=8:THEN
F$="0":F$      :rem 157
840 TS=F$:ZK=PEEK(53)+256*PEEK
(54)-LEN(T$):POKE782,ZK/25
6      :rem 2
841 POKE781,ZK-PEEK(782)*256:P
OKE780,LEN(T$):SYS65469
      :rem 107
845 POKE780,1:POKE781,DV:POKE7
82,1:SYS65466      :rem 78
850 POKE780,0:SYS65493      :rem 11
860 IF(PEEK(783)AND1)OR(19AND
ST)THEN870      :rem 111
865 PRINT"[DOWN] [DONE. *GOTO318
      :rem 96
870 PRINT"[DOWN] ERROR ON LOAD.
[2 SPACES] TRY AGAIN. [DOWN]
*IFDV=1:THEN830      :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E
2$:PRINTI$;E2$:CLOSE15:GO
TO800      :rem 102
1000 REM BUZZER      :rem 135
1001 POKE54296,15:POKE54277,45
+POKE54278,165      :rem 207
1002 POKE54276,33:POKE 54273,6
+POKE54272,5      :rem 42
1003 FORI=1:TO280:NEXT:POKE5427
6,32:POKE54273,0:POKE5427
2,0:RETURN      :rem 202
2000 REM BELL SOUND      :rem 78
2001 POKE54296,15:POKE54277,0
+POKE54278,247      :rem 152
2002 POKE 54276,17:POKE54273,4
0:POKE54272,0      :rem 86
2003 FORI=1:TO180:NEXT:POKE5427
6,16:RETURN      :rem 57
3000 PRINTC$;"[RVS] NOT ZERO PA
GE OR ROM":GOTO1800
      :rem 89

```

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a program to check your typing—"The Automatic Proofreader."

Programs for the IBM, TI-99/4A, and Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these Commodore and Atari characters in curly braces; *do not type the braces*. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

Graphics characters entered with the Commodore logo key are enclosed in a special bracket: {<A>}. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or {<8 Q>}, you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse video

Atari 400/800/XL/XE

When you see	Type	See
{CLEAR}	ESC SHIFT <	↵ Clear Screen
{UP}	ESC CTRL -	↑ Cursor Up
{DOWN}	ESC CTRL +	↓ Cursor Down
{LEFT}	ESC CTRL +	← Cursor Left
{RIGHT}	ESC CTRL +	→ Cursor Right
{BACK S}	ESC DELETE	⌫ Backspace
{DELETE}	ESC CTRL DELETE	⌫ Delete character
{INSERT}	ESC CTRL INSERT	⌫ Insert character
{DEL LINE}	ESC SHIFT DELETE	⌫ Delete line
{INS LINE}	ESC SHIFT INSERT	⌫ Insert line
{TAB}	ESC TAB	→ TAB key
{CLR TAB}	ESC CTRL TAB	⌫ Clear tab
{SET TAB}	ESC SHIFT TAB	⌫ Set tab stop
{BELL}	ESC CTRL 2	🔔 Ring buzzer
{ESC}	ESC ESC	⌫ Escape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME	⌫	{1 S}	COMMODORE 1	⌫
{HOME}	CLR/HOME	S	{2 S}	COMMODORE 2	⌫
{UP}	SHIFT ↑ CRSR	↑	{3 S}	COMMODORE 3	⌫
{DOWN}	↑ CRSR	↓	{4 S}	COMMODORE 4	⌫
{LEFT}	SHIFT ← CRSR	←	{5 S}	COMMODORE 5	⌫
{RIGHT}	← CRSR	→	{6 S}	COMMODORE 6	⌫
{RVS}	CTRL 9	R	{7 S}	COMMODORE 7	⌫
{OFF}	CTRL 0	0	{8 S}	COMMODORE 8	⌫
{BLK}	CTRL 1	1	{F1}	⌫	
{WHIT}	CTRL 2	2	{F2}	SHIFT ⌫	⌫
{RED}	CTRL 3	3	{F3}	⌫	⌫
{CYN}	CTRL 4	4	{F4}	SHIFT ⌫	⌫
{PUR}	CTRL 5	5	{F5}	⌫	⌫
{GRN}	CTRL 6	6	{F6}	SHIFT ⌫	⌫
{BLU}	CTRL 7	7	{F7}	⌫	⌫
{YEL}	CTRL 8	8	{F8}	SHIFT ⌫	⌫
				←	⌫

key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as [SPACE].

Amiga program listings contain only one special character, the left arrow (-) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN or move the cursor off the line to enter that line into memory. Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do not use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenact the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR(1536) to reenact it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a checksum.

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to re-save it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE "filename".A.

Program 1: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:REA
D A:POKE I,A:CK=CK+A:I
EXT I
120 IF CK<>19072 THEN ? "E
rror in DATA Statement
s. Check Typing.":END

130 A=USR(1536)
140 ? : ? "Automatic Proofs
reader Now Activated."
150 END
160 DATA 104,160,0,185,26,
3,201,69,240,7
170 DATA 200,200,192,34,20
8,243,96,200,169,74
180 DATA 153,26,3,200,169,
6,153,26,3,162
190 DATA 0,189,0,228,157,7
4,6,232,224,16
200 DATA 208,245,169,93,14
1,78,6,169,6,141
210 DATA 79,6,24,173,4,228
,105,1,141,95
220 DATA 6,173,5,228,105,0
,141,96,6,169
230 DATA 0,133,203,96,247,
238,125,241,93,6
240 DATA 244,241,115,241,1
24,241,76,205,238
250 DATA 0,0,0,0,0,32,62,2
46,8,201
260 DATA 155,240,13,201,32
,240,7,72,24,181
270 DATA 203,133,203,104,4
0,94,72,152,7,138
280 DATA 72,160,0,169,128,
145,88,200,192,48
290 DATA 208,249,165,203,7
4,74,74,24,105
300 DATA 161,160,3,145,88,
165,203,41,15,24
310 DATA 105,161,200,145,0
8,169,0,133,203,104
320 DATA 170,104,168,104,4
0,96
```

Program 2: IBM Proofreader

By Charles Brannon, Program Editor

```
10 "Automatic Proofreader Vers
ion 3.0 (Lines 205,286 add
ed/198 deleted/470,498 chang
ed from V2.0)
100 DIM L$(500),LNUM(500):CLO
R 0,7,KEYDEF:CLS:MAX=0:
LNUM(0)=65536:
110 ON ERROR GOTO 120:KEY 15,C
HRS(4)+CHRS(70):ON KEY(15)
GOSUB 640:KEY (15) ON:GOT
O 130
120 RESUME 130
130 DEF SEG=M40:W=PEEK(M44A)
140 ON ERROR GOTO 650:PRINT:PR
INT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN:INT
(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POK
E 1052,34:POKE 1054,0:POKE
1055,79:POKE 1056,134:POKE
1057,28:LINE INPUT L$:DEF
SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)="" THEN L$
=MID$(L$,2)GOTO 170
```



```

100 IF VAL(LEFT$(L$,2))=0 AND
MID$(L$,3,1)="" THEN L$=M
ID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'no
line number, therefore co
mand
285 BL=INSTR(L$, " ") IF BL=0 T
HEN BL=L$:GOTO 286 ELSE B
L$=LEFT$(L$,BL-1)
286 LNUM=VAL(BL$):TEXT$=MID$(L
$,LEN(STR(LNUM))+1)
290 IF TEXT$="" THEN GOSUB 540
310 IF LNUM=LNUM(P) THEN GOSU
B 560 GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$)
:CKSUM=(CKSUM+ASC(MID$(L$,
I,1)))+1 AND 255:NEXT I:LOCATE
Y,1:PRINT CHR$(65+CKSUM/1
6)+CHR$(65+(CKSUM AND 15))
+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
THEN L$(P)=TEXT$:GOTO 150
'replace line
240 GOSUB 580:GOTO 150 'insert
the line
260 TEXT$="" :FOR I=1 TO LEN(L$)
: A=ASC(MID$(L$,I,1)):TEXT$=
TEXT$+CHR$(A+32*(A%96 AND
A<123)):NEXT I
270 DELIMITR=INSTR(TEXT$, " ")
:COMMAND$=TEXT$:ARG$="" :IF
DELIMITR THEN COMMAND$=
LEFT$(TEXT$,DELIMITR-1):AR
G$=MID$(TEXT$,DELIMITR+1)
ELSE DELIMITR=INSTR(TEXT
$,CHR$(24)):IF DELIMITR T
HEN COMMAND$=LEFT$(TEXT$,D
ELIMITR-1):ARG$=MID$(TEXT
$,DELIMITR)
280 IF COMMAND$<"LIST" THEN 4
10
290 OPEN "scrn:" FOR OUTPUT AS
#1
300 IF ARG$="" THEN FIRST=0:P=
MAX-1:GOTO 340
310 DELIMITR=INSTR(ARG$,"-")
:IF DELIMITR=0 THEN LNUM=V
AL(ARG$):GOSUB 540:FIRST=P
:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIM
ITER)):LAST=VAL(MID$(ARG$,
DELIMITR+1))
330 LNUM=FIRST:GOSUB 540:FIRST
=P:LNUM=LAST:GOSUB 540:IF
P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:L$=MID$(S
TR$(LNUM(X)),2)+""
350 IF CKFLAG=0 THEN A$="" :GOT
O 370
360 CKSUM=A$+0-N$+L$(X):FOR I=
1 TO LEN(A$):CKSUM=(CKSUM+
ASC(MID$(A$,I,1)))+1 AND 255
:NEXT I:TEXT$=CHR$(65+CKSUM/16)
+CHR$(65+(CKSUM AND 15))+"
"
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<" " THEN X=P
390 NEXT X:CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LIST" THEN O
PEN "lpt1:" FOR OUTPUT AS
#1:GOTO 300
420 IF COMMAND$="CHECK" THEN C
KFLAG=1:GOTO 290
430 IF COMMAND$<"SAVE" THEN 4
50
440 GOSUB 600:OPEN ARG$ FOR OU
TPUT AS #1:ARG$="" :GOTO 30
0
450 IF COMMAND$<"LOAD" THEN 4
90

```

```

460 GOSUB 600:OPEN ARG$ FOR IN
PUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INPU
T #1,L$:BL=INSTR(L$, " ")
:BL$=LEFT$(L$,BL-1):LNUM(P)=
VAL(BL$):L$(P)=MID$(L$,LEN
(STR$(VAL(BL$))+1)):P=P+1
:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INP
UT "Erase program - Are yo
u sure?":L$:IF LEFT$(L$,1)
="Y" OR LEFT$(L$,1)="" THEN
N MAX=0:LNUM(0)=65536:GOT
O 130:ELSE 130
500 IF COMMAND$="BASIC" THEN C
OLOR 7,0,0:ON ERROR GOTO 0
:CLS:END
510 IF COMMAND$<"FILES" THEN
520
515 IF ARG$="" THEN ARG$="A:"
ELSE SEL=:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 1
30
540 P=0:WHILE LNUM=LNUM(P) AND
P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:L
NUM(X)=LNUM(X+1):L$(X)=L$(
X+1):NEXT X:RETURN
580 MAX=MAX-1:FOR X=MAX TO P+1
STEP -1:LNUM(X)=LNUM(X-1)
:L$(X)=L$(X-1):NEXT X:L$(P)=
TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34)
THEN 520 ELSE ARG$=MID$(A
RG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34)
THEN ARG$=LEFT$(ARG$,LEN(
ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$, ".
")=0 THEN ARG$=ARG$+".BAS"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"St
opped." :RETURN 150
650 PRINT "Error #":ERR:RESUME
150

```

Program 3: Commodore Proofreader

By Phillip Nelson, Assistant Editor

```

10 VEC=PEEK(772)+256*PEEK(773)
:LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADER
FOR " :IF VEC=42364 THEN
[SPACE]PRINT "C-64"
30 IF VEC=58556 THEN PRINT "VI
C-20"
40 IF VEC=35158 THEN GRAPHIC C
LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=43:HI=
46:GRAPHIC CLR:PRINT"120"
60 SA=(PEEK(LO)+256*PEEK(HI))+
6:ADR=SA
70 FOR J=2 TO 166:READ BYT:POKE
E ADR,BYT:ADR=ADR+1:CHK=CHK
+BYT:NEXT
80 IF CHK<>20570 THEN PRINT "E
RROR* CHECK TYPING IN DATA
STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
RS=SA+RF:HB=INT(RS/256):LO=
RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
F,LF:POKE SA+HF,HB:NEXT
110 IF CHK<>22054 THEN PRINT "
*ERROR* RELOAD PROGRAM AND

```

```

[SPACE]CHECK FINAL LINE":EN
D
120 POKE SA+149,PEEK(772):POKE
SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
14,22:POKE SA+18,23:POKE SA+
29,224:POKE SA+139,224
140 PRINT CHR$(147):CHR$(17):"
PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
EEK(LO)+256*PEEK(HI))-1,0:N
EW
160 DATA 128,169,73,141,4,3,16
9,3,141,5,3
170 DATA 88,96,165,20,133,167,
165,21,133,169,169
180 DATA 0,141,0,255,162,31,18
1,199,157,227,3
190 DATA 202,16,248,169,19,32,
218,255,169,18,32
200 DATA 218,255,160,0,132,180
,132,176,136,230,180
210 DATA 208,185,0,2,240,46,20
1,34,208,8,72
220 DATA 165,176,73,255,133,17
6,184,72,201,32,208
230 DATA 7,165,176,200,3,104,2
08,226,184,166,180
240 DATA 24,165,167,121,0,2,13
3,167,165,168,105
250 DATA 0,133,168,202,200,239
,240,282,165,167,69
260 DATA 168,72,41,15,160,185,
211,3,32,210,255
270 DATA 184,74,74,74,74,168,1
85,211,3,32,210
280 DATA 255,162,31,189,227,3,
149,199,202,16,248
290 DATA 169,146,32,210,255,76
,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
116,117,151,128,129,167,136
,137

```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```

10 C = 0: FOR I = 760 TO 768 +
60: READ A=C + C + A: POKE I
,A: NEXT A
20 IF C < 7258 THEN PRINT "ER
ROR IN PROOFREADER DATA STA
TEMENTS": END
30 IF PEEK (190 + 256) < 76 T
HEN POKE 56,0: POKE 57,3: CA
LL 1002: GOTO 50
40 PRINT CHR$(4):IN$=300
50 POKE 34,0: HOME : POKE 34,1:
VTAB 2: PRINT "PROOFREADER
INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 200,60,139,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 248,8,184,10,175,255
140 DATA 1,100,0,72,202,208
150 DATA 230,104,70,41,15,9
160 DATA 40,201,50,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 40,201,50,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96

```

It Talks! It Recognizes! It Writes Music! and more...



THE AMAZING VOICE MASTER® Speech and Music Processor

Your computer can talk in your own voice. Not a synthesizer but a true digitizer that records your natural voice quality—and in any language or accent. Words and phrases can be expanded without limit from disk.

And it will understand what you say. A real word recognizer for groups of 32 words or phrases with unlimited expansion from disk memory. Now you can have a two way conversation with your computer!

Easy for the beginning programmer with new BASIC commands. Machine language programs and memory locations for the more experienced software author.

Exciting Music Bonus lets you hum or whistle to write and perform. Notes literally scroll by as you hum! Your composition can be edited, saved, and printed out. You don't have to know one note from another in order to write and compose!

Based upon new technologies invented by COVOX. One low price buys you the **complete system**—even a voice controlled blackjack game! In addition, you will receive a subscription to COVOX NEWS, a periodic newsletter about speech technology, applications, new products, up-dates, and user contributions. You will never find a better value for your computer.

ONLY \$89.95 includes all hardware and software. For telephone demonstration or additional information, call (503) 342-1271. FREE audio demo tape and brochure available. Available from your dealer or by mail. When ordering by mail add \$4.00 shipping and handling (\$10.00 for foreign, \$6.00 Canada).

The Voice Master is available for the C64, C128, all Apple II's, and Atari 800, 800X, and 130XE. Specify model when ordering.



For Faster Service on Credit Card Orders only:

ORDER TOLL FREE 1-800-523-9230



COVOX INC. (503) 342-1271
675-D Conger Street, Eugene, OR 97402
Telex 706017 (AV ALARM UD)

Advertisers Index

Reader Service Number/Advertiser	Page
102 About Software	37
103 Abacus Software	39
104 Artificial Intelligence Research Group	71
105 Alsat	24-25
C O M B Direct Marketing Corp	57
Commodore	9C
106 Compuserve	1
Computer Ability	61
The Computer Book Club	27
Computer Direct	54-57
107 Computer Mail Order	14-15
Covox Inc	126
108 Duplicating Technologies, Inc.	18
109 Electronic Arts	8C
110 Electronic Arts	2-3
111 Etek-Tek, Inc.	11
112 EPYX	118
Holt Institute	118
113 JMI Music World	29
114 Kyon Software	63
Lyco Computer	62-63
115 Marlynne Publishers, Inc.	66
116 95/4A National Assistance Group	118
NES Schools	35
117 Precision Data Products	126
118 Professor Jones	126
119 Puma	19
120 Quinsell, Inc.	59
121 subGIG Corporation	18C
122 TimeWorks, Inc.	7
123 Uniflex	118
124 White House Computer	106

COMPUTE! Classifieds	120
COMPUTE! Disk Subscription	17
COMPUTE! Subscription	33
COMPUTE!'s Apple Applications Special	49
COMPUTE!'s Elementary Amiga BASIC and Elementary ST BASIC	4
COMPUTE!'s IBM Books Collection	23
COMPUTE!'s Kids and the Atari ST and Kids and the 128	64
COMPUTE!'s Programmer's Guide	43
40 Great Flight Simulator Adventures	9
126 Machine Language for Beginners	13

PROFESSIONAL HANDICAPPING SYSTEMS

• PRESENTED BY PROFESSOR JONES •

GLTD, Thoroughbred "Gold" Edition™
A full featured handicapping analysis designed for the professional and the serious horse fan.
\$159.95 complete

EGLD, Enhanced "Gold" Edition™
"Gold" Edition with complete Master Better™ system integrated into the same disc. This powerful program will transfer all horses and scores to the bet analysis with a single keystroke.
(Master Better™ included) \$199.95 complete

GLTD, Limited "Gold" Edition™
The GLTD Professional handicapping analysis... designed to give the user the best analysis with a single keystroke.
(contents integrated better™) \$299.95 complete

GD, Gold Dog Analysis™ \$149.95
The GD Professional handicapping analysis... designed to give the user the best analysis with a single keystroke.
Limited Version \$299.95

MHH, Master Hamese Handicapper™ complete \$159.95
Professional software designed to provide thoroughbred handicapping and racing news in the United States and Canada.
w/Integrated Better™ \$199.95
Limited \$299.95

Professor Pix Football™ \$49.95
Complete 1987-88 NCAA Football on disc base allowing "disposable" previous games to be evaluated.
w/Win Loss Power Ratings \$149.95

NBA, Basketball™ NBA \$99.95 w/hotline \$129.95
w/Power Ratings \$149.95

LGT, Lottery Analysis™ \$149.95
Statistical comparison program designed to direct subtle patterns in winning numbers and digits.
w/Lotto (Max 99 Digits) \$99.95

PC-3 Portable Computer (4k) w/outside of Thoroughbred Day... \$249.95 (includes portable computer & program)

M 100 Portable (32k) w/outside of Thoroughbred Day... \$49.95 (includes portable computer & program)

FREE CATALOG!

★ ★ ★ ★
"Your BEST
BET in
handicapping
software..."

Reviewed by
Whole Earth
Software Guide
Corona Data
Systems

Prof. Jones
1940 W. State
Boise, ID 83702



48 HR. FREE SHIPPING

CALL
(208) 342-6939

TERMS: Free shipping on software orders \$25.00 or more. C.O.D. or L.P. \$5.00. 5% discount on orders \$50.00 or more. 10% discount on orders \$100.00 or more. All orders subject to change.



Flight Simulator II Scenery Disks

The Challenge of Accomplished Flight

With a realism comparable to (and in some ways even surpassing) \$100,000 aircraft flight simulators, Flight Simulator II includes full flight instrumentation and avionics, and provides a full-color out-the-window view. Instruments are arranged in the format standard to modern aircraft. All the radios needed for IFR flight are included. Front, rear, left, right, and diagonal views let you look in any direction. Program features are clearly documented in a 96-page Pilot's Operating Handbook.

For training in proper flight techniques, Flight Simulator II includes another 96-page instruction manual, compiled by two professional flight instructors with over 8,000 hours flight time and 12,000 hours of aviation teaching experience. You'll learn FAA-recommended flight procedures, from basic aircraft control through instrument approaches. To reward your accomplishments, the manual even includes a section on aerobatic maneuvers.

The Realism and Beauty of Flight

Go sight-seeing over detailed, realistic United States scenery. High-speed graphic drivers provide an animated out-the-window view in either day, dusk, or night flying modes.

Flight Simulator II features over 80 airports in four different scenery areas: New York, Chicago, Seattle, and Los Angeles. Six additional Scenery Disks covering the entire Western half of the United States are now available in IBM and C64/128 disk formats.

Apple and Atari versions will be released soon. Each disk covers a geographical region of the country in detail, and is very reasonably priced.

The Pure Fun of "World War I Ace"

When you think you're ready, you can test your flying skills with the "World War I Ace" aerial battle game. This game sends you on a bombing run over heavily-defended enemy territory. Six enemy fighters will attempt to engage you in combat as soon as war is declared. Your aircraft can carry five bombs, and your machine guns are loaded with 100 rounds of ammunition.

See Your Dealer. Flight Simulator II is available on disk for the Apple II, Atari XUXE, and Commodore 64/128 computers for \$49.95. Scenery Disks for the C64 and IBM PC (Jet or Microsoft Flight Simulator) are \$19.95 each. A complete Western U.S. Scenery six-disk set is also available for \$99.95. For additional product or ordering information, call (800) 637-4983.

Apple II is a trademark of Apple Computer, Inc.
Atari XL and XE are trademarks of Atari Corp.
Commodore 64 and 128 are trademarks of Commodore Electronics Ltd.
IBM PC is a registered trademark of International Business Machines Corp.

subLOGIC
Corporation
713 Edgemoor Drive
Champaign IL 61820
(217) 359-6462 Telex 206996

Order Line: (800) 637-4983
(except Illinois, Alaska and Hawaii)



All you need to do this



graph a spreadsheet



write a novel



fix an engine



compose a song



paint a picture



your banking



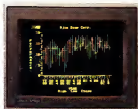
learn to fly



organize a data base



tell a story



forecast sales



is this.



When it comes to personal computers, you want the smartest, at a price that makes sense.

The new Commodore 128™ system has a powerful 128K memory, expandable by 512K. An 80-column display and 64, 128 and CP/M® modes for easy access to thousands of educational, business and home programs. And a keyboard, with built-in numeric keypad, that operates with little effort.

Or if the Commodore 128 is more machine than you had in mind, you can pick up the Commodore 64®. The Commodore 64 is our lower-priced model geared to more fundamental, basic needs.

Discover personal computers that do more for you. At prices you've been waiting for. From the company that sells more personal computers than IBM® or Apple®.

© 1985 Commodore Electronics Limited
 ® CPM is a registered trademark of Digital Research, Inc.
 ® Apple is a registered trademark of Apple Computer, Inc.
 ® IBM is a registered trademark of International Business Machines Corporation
 ® Commodore 64 is a registered trademark of Commodore Electronics, Ltd.

COMMODORE 128 AND 64 PERSONAL COMPUTERS

A Higher Intelligence

RETROMAGS

Our goal is to preserve classic video game magazines so that they are not lost permanently.

People interested in helping out in any capacity,
please visit us at www.retromags.com.

No profit is made from these scans, nor do we offer anything
available from the publishers themselves.

If you come across anyone selling releases from
this site, please do not support them and do let us know.

Thank you!

